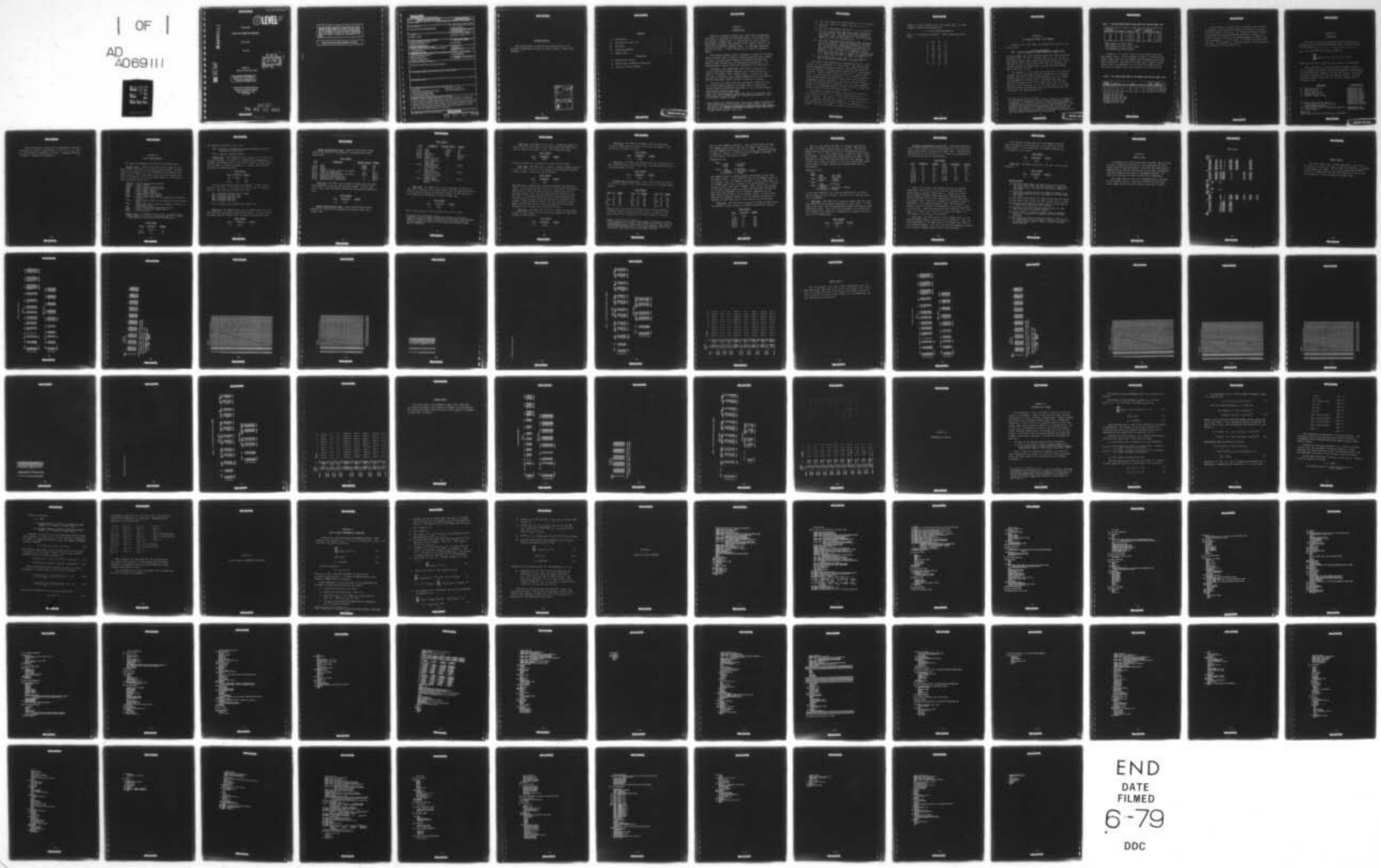
AD-A069 111 INSTITUTE FOR DEFENSE ANALYSES ARLINGTON VA SYSTEMS E--ETC F/G 15/6
THE ALLOT COMPUTER PROGRAM.(U)
APR 79 J A ROSS MDA903-79-C-0018
UNCLASSIFIED IDA-N-854 IDA/HQ-79-21150 NL

1 OF 1
AD
A069111



END
DATE
FILMED
6-79
DDC

AD-E500 065

UNCLASSIFIED

LEVEL III

AD A069111

IDA ~~Note~~ N-854

THE ALLOT COMPUTER PROGRAM

James A. Ross

April 1979

DDC
RECEIVED
MAY 30 1979
B

Prepared for

Office of the Joint Chiefs of Staff

DDC FILE COPY

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

INSTITUTE FOR DEFENSE ANALYSES
SYSTEMS EVALUATION DIVISION

Contract MDA903 79 C 0018
Task T-8-041

IDA Log No. HQ 79-21150
Copy 47 of 50 Copies

79 05 11 052

UNCLASSIFIED

403307

The work reported in this document was conducted under contract MDA903 79 C 0018 OUSDRE for the Office of the Joint Chiefs of Staff. The publication of this IDA Note does not indicate endorsement by the Office of the Joint Chiefs of Staff nor should the contents be construed as reflecting the official position of that agency.

Approved for public release; distribution unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) THE ALLOT COMPUTER PROGRAM		5. TYPE OF REPORT & PERIOD COVERED FINAL REPORT
		6. PERFORMING ORG. REPORT NUMBER IDA Note N-854
7. AUTHOR(s) JAMES A. ROSS		8. CONTRACT OR GRANT NUMBER(s) MDA903 79 C 0018 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS Institute for Defense Analyses 400 Army-Navy Drive Arlington, Virginia 22202		10. PROGRAM ELEMENT PROJECT TASK AREA & WORK UNIT NUMBERS Task T-8-041
11. CONTROLLING OFFICE NAME AND ADDRESS Office of the Under Secretary of Defense for Research and Engineering Washington, D.C. 20301		12. REPORT DATE April 1979
		13. NUMBER OF PAGES 91
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) DoD-IDA Management Office 400 Army-Navy Drive Arlington, Virginia 22202		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Target value
nuclear target vulnerability (VN) system
equivalent megatonnage (EMT)
countermilitary potential (CMP)
relative force size (RFS)
circular error probable (CEP)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ALLOT is a computer program that uses linear programming techniques to determine the optimum laydown of a nuclear arsenal against a target base. The arsenal (on input to the model) consists of a number of weapon classes, each characterized by yield, burst height, CEP, prelaunch survivability, weapon system reliability, and penetration probability. The target base (also an input) consists of a number of target classes, each characterized by a target value, a target vulnerability number, and a target base.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

79 05 11 052

ACKNOWLEDGEMENT

The author wishes to express his appreciation for the invaluable assistance provided by LtCol. Donald J. Berg, USAF, JCS/SAGA.

CONTENTS

Chapter I

INTRODUCTION

ALLOT is a computer program that uses linear programming techniques to determine the optimum laydown of a nuclear arsenal against a target base. It was written by LtCol. Donald J. Berg, USAF, JCS/SAGA, and modified by Dr. James A. Ross of the Institute for Defense Analyses (IDA). It has been validated against the SIOP. A brief description of ALLOT is presented below.

The nuclear arsenal (input) comprises a number of different weapon classes, each of which is characterized by yield, height of burst, circular error probable, prelaunch survivability, weapon system reliability, and penetration probability. All weapons within a given class are considered to be identical.

Similarly, the target base (also input) consists of a number of different target classes, each characterized by a value, a VN number,[1] and a target radius. In addition, the user may, if desired, input minimum damage goals to be achieved against individual target classes, or groups of classes.

Once all the inputs have been assembled, ALLOT allocates the nuclear arsenal against the target base to produce an optimum laydown according to whichever of the following specifications is chosen by the user.

NOTE: The views expressed herein are those of the author only. Publication of this Note does not indicate endorsement by IDA or the Joint Chiefs of Staff.

[1] Those readers not familiar with the nuclear target vulnerability (VN) system used by the Defense Intelligence Agency (DIA) are referred to the Physical Vulnerability Handbook--Nuclear Weapons, DIA publication AP-550-1-2-69-INT, published 1 June 1969, with changes dated through 1 June 1976.

(1) *The total target value destroyed is to be a maximum.*

(2) *The total number of weapons employed is to be a minimum.*

(3) *The total megatonnage employed is to be a minimum.*

(4) *The total equivalent megatonnage (EMT) employed is to be a minimum.* The EMT of a weapon is defined as follows. Let Y be the yield of the weapon, in megatons. Then the EMT equals $Y^{1/2}$ if Y is greater than 1, or $Y^{2/3}$ if Y is less than 1.

(5) *The total countermilitary potential (CMP) employed is to be a minimum.* For the purposes of the ALLOT model, the CMP of a weapon is defined as follows. Let Y be the yield of the weapon, in megatons, and let CEP be the circular error probable, in feet. Define $C = 10^6 \cdot Y^{2/3}/CEP^2$ if Y is greater than or equal to 0.2, or $C = 10^6 \cdot Y^{4/5}/CEP^2$ if Y is less than 0.2. Then the CMP equals C unless C is greater than 3, in which case the CMP is simply set equal to 3.

(6) *The Relative Force Size (RFS) is to be computed.* RFS is defined and discussed in Chapter II.

The chief ALLOT modification performed by IDA is the inclusion of the capability to compute RFS. This measure has gained widespread interest recently, and has been used both in the Secretary of Defense Annual Report and the Consolidated Guidance. ALLOT formerly had no capability to compute RFS; hence, its inclusion is considered to be a valuable addition. Other modifications performed by IDA, aside from auxiliary programming changes required to facilitate the RFS computation within the existing ALLOT framework, include coding changes designed to accommodate greater numbers of weapon classes and target classes than was previously possible.

The version of ALLOT currently being used on IDA's CDC 6400 computer allows up to 20 weapon classes and up to 20 target classes. (The version in JCS/SAGA allows considerably more.) Computer running time depends to a large extent on n, the maximum number of weapons allowed to be assigned to a target. If NW is the number of weapon classes, and NT is the

number of target classes, then the running time t on IDA's computer is given approximately by

$$t = 0.001\ (NW)^{n}(NT)(NW+NT/n!$$

where t is expressed in minutes. Some examples are given below.

n	NW	NT	t
1	5	5	<0.5
1	10	10	2
1	15	15	7
1	20	20	16
2	5	5	1
2	10	5	4
2	10	10	10
2	15	10	28
2	15	15	50
3	5	5	1
3	10	5	12
3	10	10	33

Chapter II

THE RELATIVE FORCE SIZE MEASURE

Relative Force Size (RFS) is defined as the ratio of two force levels:

$$\text{RFS} = \frac{\text{forces available}}{\text{forces required to meet specified damage goals}}$$

In other words, the forces required equal the forces available divided by RFS. To put it another way, if all the forces in the weapon arsenal were scaled down proportionally by a factor RFS, and if these scaled forces were then all optimally allocated against the target base, then the specified damage goals would just be met.

For example, if the target base consisted of 50 identical targets, and if the weapon arsenal consisted of 50 identical weapons, each with a single-shot probability of kill (SSPK)[1] of 0.50 against a target, and if the damage goal were to destroy 50 percent of the total target base, then the RFS would be 1.00 (see Table 1). On the other hand, if the goal were 25 percent, the RFS would be 2.00; and if the goal were 75 percent, the RFS would be 0.50.

[1]For the purposes of this paper, SSPK is defined as (probability of arrival) x (probability of kill given an arrival), where probability of arrival is defined as (prelaunch survivability) x (weapon system reliability) x (penetration probability). All warheads are assumed to be independent, and no fratricide is allowed. The effect of an enemy attack absorbed before the force is launched may be reflected in prelaunch survivabilities, but post-attack retargeting is not permitted.

Table 1. RFS CALCULATIONS BASED ON ONE WEAPON TYPE AND ONE TARGET TYPE

Weapons Available	Targets	SSPK	Overall Goal	Weapons Required	RFS
50	50	0.50	50%	50[a]	1.00
50	50	0.50	25%	25[b]	2.00
50	50	0.50	75%	100[c]	0.50

[a]One weapon on every target.
[b]One weapon on every other target.
[c]Two weapons on every target.

In general, of course, there are more target types and more weapon types. In the examples in Table 2, there are two target types (each with 50 targets) and two weapon types (each with 50 weapons). The SSPKs for weapon type 1 against target types 1 and 2 are assumed to be 0.10 and 0.90 respectively, and those for weapon type 2 are assumed to be 0.50 and 0.50.

Table 2. RFS CALCULATIONS BASED ON TWO WEAPON TYPES AND TWO TARGET TYPES

Weapons Available		Targets		SSPKs				Overall Goals		Weapons Required		RFS
Type 1	Type 2	Type 1	Type 2	Wpn 1/Tgt 1	Wpn 1/Tgt 2	Wpn 2/Tgt 1	Wpn 2/Tgt 2	Tgt 1	Tgt 2	Type 1	Type 2	
50	50	50	50	0.10	0.90	0.50	0.50	0.50	0.90	50[a]	50[b]	1.00
50	50	50	50	0.10	0.90	0.50	0.50	0.25	0.45	25[c]	25[d]	2.00
50	50	50	50	0.10	0.90	0.50	0.50	0.75	0.99	100[e]	100[f]	0.50

[a]One weapon on every type 2 target.
[b]One weapon on every type 1 target.
[c]One weapon on every other type 2 target.
[d]One weapon on every other type 1 target.
[e]Two weapons on every type 2 target.
[f]Two weapons on every type 1 target.

As the numbers of target types and weapon types continue to increase, it becomes more and more difficult to evaluate the RFS by simple inspection. The general problem is amenable to solution by linear programming techniques, however, provided a bound is placed on the maximum number of weapons that can be assigned to any one target. In ALLOT this upper bound is three. For problems of practical interest, this limitation has not been a serious restriction.

Chapter III

OPERATION

The types of problems solved by ALLOT can always be expressed in the following mathematical form: Find the values of the variables x_1, x_2, x_3, ..., x_{m-1}, x_m; y_1, y_2, y_3, ..., y_{n-1}, y_n, such that, for given a_{ij} and b_i,

$$\sum_{j=1}^{n} a_{ij}\, y_j + x_i = b_i \quad (i = 1,2,\ldots,m),$$

where $x_i \geq 0$ for all i, $y_j \geq 0$ for all j, and x_1 is a minimum.

Thus, *operation of the ALLOT model* in essence consists of (1) reading the inputs, (2) making preliminary calculations, (3) determining the values of the a_{ij} and the b_i, (4) solving the basic equations, and (5) printing the results. The subroutines in which these steps are actually carried out are indicated below.

OPERATION	SUBROUTINE(S)
1. Read Input Values	SUBROUTINE INPUT
2. Compute SSPKs	SUBROUTINE DPDX
3. Print Input Values	SUBROUTINE INPUT
4. Compute Values of b_i	SUBROUTINE INPUT
5. Compute Values of a_{ij}	SUBROUTINE BUILDA
	SUBROUTINE CREATE
	SUBROUTINE INSERT
	FUNCTION OBJECT
6.* Print Values of a_{ij} and b_i	SUBROUTINE OUTMAT
7. Solve Equations by Using Linear Programming Algorithm	SUBROUTINE SIMPLE**
8.* Print Formal Linear Programming Solution	SUBROUTINE DUAL
9. Print Results	SUBROUTINE OUTPUT

*Optional
**Also uses auxiliary subroutines AX and AMAT.

Further discussion concerning the mathematics involved--including the physical significance of the x_i, y_j, a_{ij}, and b_i--is presented in Appendices A and B. A complete listing of ALLOT is presented in Appendix C.

Chapter IV

INPUT REQUIREMENTS

The input requirements for ALLOT are described below.

Control Cards - Control cards are used to signal to the input routine the type of information that follows. In some cases the information is fully contained on the control card itself, and in other cases the control card only denotes the type of information on the cards that follow it. The control cards (discussed below) are:

*IN/OUT - select input/output options
*START - start problem
*WPN - input weapon specifications
*TGT - input target specifications
*NWP - input weapon numbers
*NTG - input target numbers
*CWP - change and/or add weapons
*CTG - change and/or add targets
*ADD - input weapon vectors to be added (see discussion below)
*SUB - input weapon vectors to be deleted (see discussion below)
*FOR - input applicable targets for weapon vectors (see discussion below)
*FTP - input footprinting degradation factor
*COL - include collateral damage option
*END - end of problem

*IN/OUT Card - The *IN/OUT card is used to select input/output options. Its use is optional. When it is used, it must appear directly before any *START card.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-7	*IN/OUT	--
10-12	f_1	A3
14-16	f_2	A3

The possible values for the f_1 are:

MAT - print out intermediate results (generally used for debugging purposes only)

NOL - suppress laydown printout

*START Card - The *START card must appear as the first card of every problem. In addition to signaling the beginning of a problem, the *START card is also used to select an upper limit on the number of weapons assigned to a target, and to select the type of laydown desired.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-6	*START	--
8	N	A1
10-12	F	A3

N is the upper weapon limit per target. It must not be greater than three. F is used to select the type of laydown desired. The possible "values" for F are:

DE = maximize expected value damage
WPN = minimize weapons used
MT = minimize megatons used
EMT = minimize EMT used
CMP = minimize CMP used

If F is blank, the program will compute RFS.

*WPN Card - The *WPN card is used to signal that the cards that follow it are weapon specification cards. If this card is not used, then the arsenal from the previous problem is used.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-4	*WPN	--

Weapon Specification Card - Weapon specification cards specify the values of the parameters associated with each weapon type.

CARD FORMAT

COLS	PARAMETER	DEFAULT VALUE	FORMAT
5-12	name	blank	2A4
15-18	type	blank	A4
19-24	number	none	I6
29	height of burst (gnd=0, opt air=1)	1	I1
31-40	yield (in megatons)	none	F10.0
41-50	circular error probable (in feet)	none	F10.0
51-60	prelaunch survivability	1	F10.0
61-70	weapon system reliability	1	F10.0
71-80	probability to penetrate	1	F10.0

*TGT Card - The *TGT card is used to signal that the cards that follow it are target specification cards. If this card is not used, then the target system from the previous problem is used.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-4	*TGT	--

Target Specification Card - Target specification cards specify the values of the parameters associated with each target type.

CARD FORMAT

COLS	PARAMETER	DEFAULT VALUE	FORMAT
5-12	name	blank	2A4
15-18	type	blank	A4
19-24	number	none	I6
26-29	VN number	none	I2,A1,I1
31-40	value	1	F10.0
41-50	DEMIN--minimum DE allowed on each target of that type.	0	F10.0
51-60	DEMAX--maximum DE allowed on each target of that type.	1	F10.0
61-70	DEAVE--minimum average DE to be achieved on that target type.[1]	0	F10.0
71-80	DEDLT--target radius (nautical miles)	0	F10.0

*NWP Card - The *NWP card is used to change the number of each name of weapon used in the arsenal from the previous problem. If the previous problem had NW weapon names, the NW numbers in 20I4 format must immediately follow the *NWP card.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-4	*NWP	--

This is followed by a card with NW items in 20I4 format.

[1]A series of n consecutive target specification cards with negative values of DEAVE, followed by a card with a positive DEAVE value, D, indicates that a fraction D of the total value of all the targets in the set of n+1 target specification cards must be destroyed.

*NTG Card - The *NTG card is used to change the number of each of the NT target names in the same manner the *NWP card is used to change the number of each of the NW weapon names.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-4	*NTG	--

This is followed by a card with NT items in 20I4 format.

*CWP Card - The *CWP card is used to change a weapon specification or add a new one to the arsenal of the previous problem, without having to input the entire arsenal.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-4	*CWP	--

The *CWP card is immediately followed by weapon specification cards. If the name and type field on a card (other than blanks) agree with the name and type field for a weapon type in the previous arsenal, then the old parameters are replaced by the new ones. If the name and type fields do not agree with any in the previous arsenal, then the weapon type is added to the arsenal. When a weapon is added, its type field must either agree with the type field of the last weapon in the previous arsenal or else it must be an entirely new type.

*CTG Card - The *CTG card is used to change or add targets to the previous target system in the same manner the *CWP card changes or adds weapons.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-4	*CTG	--

*ADD Card - The *ADD card signals that the cards that follow it define weapon vectors[1] that are to be added to the allowed set of weapon vectors.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-4	*ADD	--

*SUB Card - The *SUB card signals that the cards that follow it define weapon vectors that are to be deleted from the set of allowed weapon vectors.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-4	*SUB	--

ADD/SUB Specification Card - These cards define the weapon vectors to be added to or deleted from the set of allowed weapon vectors.

CARD FORMAT

COLS	PAR	FORMAT	COLS	PAR	FORMAT	COLS	PAR	FORMAT
6	r_1	A1	30	r_3	A1	54	r_5	A1
7	n_1	I1	31	n_3	I1	55	n_5	I1
9-16	t_1	2A4	33-40	t_3	2A4	57-64	t_5	2A4
18	r_2	A1	42	r_4	A1	66	r_6	A1
19	n_2	I1	43	n_4	I1	67	n_6	I1
21-28	t_2	2A4	45-52	t_4	2A4	69-76	t_6	2A4

The r_i's are only used in the *SUB section and are relational operators, where G signifies greater than, L signifies less than, and any other character or a blank signifies equality.

[1]Weapon vectors are, by definition, weapon combinations allowed to be assigned to a target. The *START card indicates that all combinations with N or less weapons are allowed. The purpose of the *ADD and *SUB cards is to permit additions and/or subtractions to the group of allowed weapon vectors.

The n_i's are numbers of weapons. The t_i's are either the name of a weapon (name field), the type of weapon (type field), or the word OTHERS left justified in the field. Anything following a blank, an invalid t_i field, or the word OTHERS is ignored. A t_6 field may be used only for OTHERS.

Examples of use:

```
*ADD
      1 ICBM        1 B52 SRAM
      2 SLBM        1 OTHERS

*SUB  1 ICBM       G0 TRIDENT
      1 POLARIS     1 B52 SRAM   0 OTHERS
     G0 POSEIDON   L1 OTHERS
```

The OTHERS clause is handled somewhat differently in the *ADD and *SUB sections. In the *ADD section, if the OTHERS clause is not present, then zero OTHERS is assumed. On the other hand, if the OTHERS clause is not present in the *SUB section, it is interpreted as "don't care about" OTHERS. Thus, in the first example of the *SUB section above, all weapon vectors with one ICBM and more than zero TRIDENTs will be deleted, regardless of how many other weapons may be in the weapon vector. In the second *SUB section example, however, only the weapon vector with one POLARIS, one B52 SRAM, and no other weapons will be deleted.

<u>*FOR Card</u> - The *FOR card is used in an *ADD or *SUB section to indicate to which targets an ADD/SUB specification applies.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-4	*FOR	--
9-16	t_1	2A4
21-28	t_2	2A4
33-40	t_3	2A4
45-52	t_4	2A4
57-64	t_5	2A4
69-76	t_6	2A4

The t_1's are either the name of a target (name field), the type of a target (type field), or ALL EXC. The ALL EXC clause can only be used in t_1, and is interpreted as for all except $t_2, \ldots, t_6$. All fields after a blank or invalid field are ignored. All specification cards between the *ADD or *SUB card and the first *FOR card are applied to all targets. Once a *FOR card is used, it applies to all the specification cards that follow it in that section until another *FOR card is encountered.

Examples of use:

```
*ADD
*FOR      HARD            NUC STOR
        1 ICBM          1 B52 SRAM
*FOR      ALL EXC         SS-18
        2 SLBM          1 OTHERS
*SUB
        1 ICBM         GO TRIDENT
*FOR      AIRFIELD
        1 POLARIS    1 B52 SRAM   0 OTHERS
       GO POSEIDON  L1 OTHERS
```

In the above examples, the first specification in the *SUB section applies to all targets, and both of the last two apply only to AIRFIELD.

*FTP Card - The *FTP card is used to signal that the cards that follow it are footprint degradation factor cards. If this card is not used, then the degradation factors from the previous problem are used. Until the first employment of an *FTP card, it is assumed that there is no degradation in weapon effectiveness due to footprinting.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-4	*FTP	--

Footprint Degradation Factor Card - Footprint degradation factor cards indicate relative effectiveness of the specified weapon name against the specified target name. (The relative effectiveness is by definition 1.0 if there is no degradation in effectiveness due to footprinting.)

CARD FORMAT

COLS	PARAMETER	FORMAT	COLS	PARAMETER	FORMAT
1-2	t_1	I2	27-28	w_4	I2
3-4	w_1	I2	29-32	f_4	F4.0
5-8	f_1	F4.0	33-34	t_5	I2
9-10	t_2	I2	35-36	w_5	I2
11-12	w_2	I2	37-40	f_5	F4.0
13-16	f_2	F4.0	...	...	...
17-18	t_3	I2	73-74	t_{10}	I2
19-20	w_3	I2	75-76	w_{10}	I2
21-24	f_3	F4.0	77-80	f_{10}	F4.0
25-26	t_4	I2			

Each t_i is a target name number, each w_i is a weapon name number, and each f_i is the relative effectiveness of weapon name w_i against target name t_i. There is no limit to the number of weapon name/target name combinations permitted, although only 10 such combinations can be indicated on a single card. However, any number of successive cards is permitted. If n weapon name/target name combinations are desired, then the columns corresponding to t_{n+1} and w_{n+1} must be left blank. (This may necessitate the inclusion of a blank card. For example, if n equals 10, blanks in columns corresponding to t_{11} and w_{11} are required.)

*COL Card - The *COL card is used to signal that the collateral damage feature in ALLOT is to be activated. Until the first employment of the *COL card, it is assumed that there is no collateral damage. Once this card has been employed, however, the collateral damage feature cannot be deactivated.

This feature assumes that the total damage resulting from a detonation is proportional to the cube root of the weapon yield. It occurs whenever the target attacked has a VN number of 15 or less.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-4	*COL	--

END Card - The *END card denotes the end of the problem statement.

CARD FORMAT

COLS	PARAMETER	FORMAT
1-4	*END	--

Miscellaneous

(a) The *WPN, *TGT, *CWP, and *CTG section must precede all *ADD, *SUB, *FTP, and *COL sections. Other than this restriction, the various sections may appear in any order.

(b) Internally, weapon vectors are added according to the ADD cards before deletions according to the SUB cards take place.

(c) The name fields within the *WPN section must be unique.

(d) The name fields within the *TGT section must be unique.

(e) The type field within a section cannot be the same as a name field within that same section.

(f) All WPN or TGT specification cards having the same type field must be grouped together.

(g) If the number of target or weapon types is changed, then it is necessary to redefine the *ADD and *SUB sections.

(h) Null *ADD and *SUB sections are allowed. That is, it is permissible to have an *ADD or *SUB card followed by no specification cards in order to delete previous specifications.

Chapter V

SAMPLE CASES

To indicate the form of the output of ALLOT, and to assist the user in employing the model, a run consisting of three sample cases is presented below. The first two cases concern computations of RFS and include the two optional intermediate printouts normally used only for debugging purposes. The third case concerns a problem in which DE is maximized.

The input deck for the run is displayed below, followed by the complete output. The total running time for these three sample cases on the IDA computer was about a minute.

INPUT DECK

```
*IN/OUT  MAT
*START 1
*WPN
     WPN1     ICBM   150   1      3.000    4000.                  0.95
     WPN2     ICBM   150   1      2.500    3000.                  0.85
     WPN3     ICBM   150   1      2.000    2000.                  0.75
     WPN4     ICBM   150   1      1.500    1000.                  0.85
     WPN5     SLBM   150   0      1.000    1500.                  0.95
     WPN6     SLBM   150   0      0.900    2500.                  0.85
     WPN7     SLBM   150   0      0.700    3500.                  0.75
     WPN8     BMBR   150   0      0.500    3000.                  0.85
     WPN9     BMBR   150   1      0.300    2000.                  0.95
     WPN10    BMBR   150   1      0.100    1000.                  0.75
*TGT
     TGT1     OBJ1   100 10G5        1.              0.8      0.70
     TGT2     OBJ1   100 12G6        1.              0.8      0.00
     TGT3     OBJ2   100 8P4         1.              0.8      0.00
     TGT4     OBJ3   100 45P2        1.              0.9      0.50
     TGT5     OBJ3   100 16G8        1.              0.9      0.65
     TGT6     OBJ3   100 40P5        1.              0.9      0.60
     TGT7     OBJ4   100 20G2        1.              0.8      1.0
     TGT8     OBJ5   100 11P1        1.              0.9      0.70
*ADD
*FOR     OBJ3
      1 SLBM      1 ICBM
*END
*START 2
*ADD
*SUB
*FOR     OBJ3
      1 SLBM      1 ICBM
*END
*IN/OUT
*START 2 DE
*WPN
     WPN1            700   1      5.000    3000.     0.9      0.50     1.0
     WPN2            500   1      3.000    2500.     0.9      0.60     0.9
     WPN3            300   1      1.000    2000.     0.9      0.70     0.8
     WPN4            500   1      0.500    1500.     0.9      0.60     0.7
     WPN5            700   1      0.200    1000.     0.9      0.50     0.6
*TGT
     TGT1             10 10G5     1.000
     TGT2             20 10G5     0.500
     TGT3             25 10G5     0.400
     TGT4             40 10G5     0.250
     TGT5             50 10G5     0.200
     TGT6            100 10G5     0.100
     TGT7            200 10G5     0.050
     TGT8            250 10G5     0.040
     TGT9            500 10G5     0.020
     TGT10          1000 10G5     0.010
*SUB
*COL
*END
```

SAMPLE CASE 1

The first sample case, an RFS computation, involves an allocation in which most target classes are limited to a maximum of one weapon per target. An ADD card allows OBJ3 targets to receive one SLBM weapon and one ICBM weapon, but this is the only exception to the one-weapon-per-target limit.

WEAPON ALLOCATION PROBLEM 1

ARSENAL

	NAME	TYPE	NUMBER	HOB	YIELD	CEP	PLS	WSR	PTP	EMT	CMP	N MT	N EMT	N CMP
1	WPN1	ICBM	150	A	3.000	4000	1.000	.950	1.000	1.732	.130	450.000	259.808	19.501
2	WPN2	ICBM	150	A	2.500	3000	1.000	.850	1.000	1.581	.205	375.000	237.171	30.700
3	WPN3	ICBM	150	A	2.000	2000	1.000	.750	1.000	1.414	.397	300.000	212.132	59.528
4	WPN4	ICBM	150	A	1.500	1000	1.000	.850	1.000	1.225	1.310	225.000	183.712	196.556
5	WPN5	SLBM	150	G	1.000	1500	1.000	.950	1.000	1.000	.444	150.000	150.000	66.667
6	WPN6	SLBM	150	G	.900	2500	1.000	.850	1.000	.932	.149	135.000	139.825	22.372
7	WPN7	SLBM	150	G	.700	3500	1.000	.750	1.000	.788	.064	105.000	118.256	9.654
8	WPN8	BMBR	150	G	.500	3000	1.000	.850	1.000	.630	.070	75.000	94.494	10.499
9	WPN9	BMBR	150	A	.300	2000	1.000	.950	1.000	.448	.112	45.000	67.221	16.805
10	WPN10	BMBR	150	A	.100	1000	1.000	.750	1.000	.215	.158	15.000	32.317	23.773
	TOTALS =		1500									1875.000	1494.935	456.054

TARGET SYSTEM

	NAME	TYPE	NUMBER	VNTK	VALUE	N VALUE	MIN DE	MAX DE	AVE DE	RADIUS
1	TGT1	OBJ1	100	10Q5	1.00	100.00	0.0000	.8000	.7000	0.0000
2	TGT2	OBJ1	100	12Q6	1.00	100.00	0.0000	.8000	0.0000	0.0000
3	TGT3	OBJ2	100	8P4	1.00	100.00	0.0000	.8000	0.0000	0.0000
4	TGT4	OBJ3	100	45P2	1.00	100.00	0.0000	.9000	.5000	0.0000
5	TGT5	OBJ3	100	16Q8	1.00	100.00	0.0000	.9000	.6500	0.0000
6	TGT6	OBJ3	100	40P5	1.00	100.00	0.0000	.9000	.6000	0.0000
7	TGT7	OBJ4	100	20Q2	1.00	100.00	0.0000	.8000	-1.0000	0.0000
8	TGT8	OBJ5	100	11P1	1.00	100.00	0.0000	.9000	.7000	0.0000
	TOTALS =		800			800.00				

UNCLASSIFIED

UF TABLE

TARGET TYPE	WEAPON TYPE 1	2	3	4	5	6	7	8	9	10
1	.950000	.850000	.750000	.850000	.950000	.849990	.746035	.840037	.950000	.750000
2	.949998	.850000	.750000	.850000	.950000	.849704	.736094	.830949	.949924	.750000
3	.950000	.850000	.750000	.850000	.950000	.850000	.749992	.849996	.950000	.750000
4	.102785	.139463	.222273	.570000	.746889	.000581	.034006	.041096	.006544	.122309
5	.948746	.849060	.750000	.850000	.949990	.842779	.691034	.788083	.942241	.749401
6	.215484	.283804	.403517	.757404	.483493	.179602	.071907	.086612	.172774	.227866
7	.812977	.823821	.747531	.850000	.930613	.735664	.483464	.568793	.847318	.727743
8	.950000	.850000	.750000	.850000	.950000	.849900	.743792	.845039	.950000	.750000

THE SOLUTION IS LIMITED TO THE FOLLOWING WEAPON VECTORS

ALL POSSIBLE WEAPON VECTORS WITH 1 OR LESS WEAPONS

PLUS ALL VECTORS OF THE FOLLOWING TYPES

FOR THE FOLLOWING TARGETS ONLY 0RJ3 .

1 ALL POSSIBLE WEAPON VECTORS WITH

1 SLBM WEAPONS

AND 1 ICBM WEAPONS

UNCLASSIFIED

C A MATRIX

	C	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8					
1	0.00	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
2	0.00	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0.00	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
4	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
5	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
7	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
8	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
9	0.00	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
10	0.00	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
11	0.00	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
12	0.00	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
13	0.00	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
14	0.00	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
15	0.00	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
16	0.00	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
17	0.00	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
18	0.00	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
19	0.00	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
20	0.00	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
21	0.00	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
22	0.00	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
23	0.00	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
24	0.00	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
25	0.00	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
26	0.00	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
27	0.00	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
28	0.00	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
29	0.00	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
30	0.00	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
31	0.00	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
32	0.00	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
33	0.00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
34	0.00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
35	0.00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
36	0.00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
37	0.00	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
38	0.00	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
39	0.00	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
40	0.00	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
41	0.00	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
42	0.00	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
43	0.00	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
44	0.00	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
45	0.00	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
46	0.00	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
47	0.00	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
48	0.00	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
49	0.00	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
50	0.00	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
51	0.00	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
52	0.00	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
53	0.00	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
54	0.00	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
55	0.00	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
56	0.00	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

UNCLASSIFIED

UNCLASSIFIED

C A MATRIX

	C	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8					
57	0.00	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
58	0.00	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
59	0.00	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
60	0.00	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
61	0.00	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
62	0.00	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
63	0.00	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
64	0.00	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
65	0.00	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
66	0.00	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
67	0.00	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
68	0.00	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
69	0.00	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
70	0.00	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
71	0.00	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
72	0.00	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
73	0.00	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
74	0.00	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
75	0.00	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
76	0.00	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
77	0.00	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
78	0.00	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
79	0.00	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
80	0.00	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
81	0.00	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
82	0.00	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
83	0.00	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84	0.00	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	0.00	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
86	0.00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
87	0.00	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
88	0.00	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
89	0.00	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
90	0.00	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
91	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
92	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
93	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
94	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
95	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
B =		150	150	150	150	150	150	150	150	150	150	100	100	100	100	100	100	100	100	070	050	065	000	140

1	81	29.040638
2	54	.072729
3	68	27.472439
4	39	76.546879
5	15	59.355782
6	35	16.562554
7	29	76.546879
8	46	10.302308
9	89	76.546879
10	9	17.191097
11	96	.510313
12	2	100.000000
13	3	100.000000
14	80	46.006241
15	36	23.380392
16	7	24.081664
17	65	10.982614
18	23	76.546879
19	1	6.262023
20	66	25.621320
21	67	2.168199
22	5	23.525850
23	52	76.474150
24	97	.510313

THE RELATIVE FORCE SIZE IS 1.96

PROBLEM 1 OPTIMAL SOLUTION FOR MAXIMUM VALUE DAMAGED

TARGET RESULTS

	TARGET	TYPE	VALUE DAMAGED	TARGETS DAMAGED NUMBER	TARGETS DAMAGED PERCENT	WEAPONS USED NUMBER	WEAPONS USED PERCENT	MEGATONS USED NUMBER	MEGATONS USED PERCENT	EMT USED NUMBER	EMT USED PERCENT	CMP USED NUMBER	CMP USED PERCENT
1	TGT1	OBJ1	70.00	70.00	70.00	93.74	6.25	55.30	2.95	64.05	4.28	7.65	1.68
2	TGT2	OBJ1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	TGT3	OBJ2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	TGT4	OBJ3	50.00	50.00	50.00	113.15	7.54	173.78	9.27	139.32	9.32	112.21	24.61
5	TGT5	OBJ3	65.00	65.00	65.00	76.47	5.10	191.19	10.20	120.92	8.09	15.65	3.43
6	TGT6	OBJ3	60.00	60.00	60.00	200.00	13.33	344.34	18.37	254.59	17.03	64.05	14.04
7	TGT7	OBJ4	55.38	55.38	55.38	75.92	5.06	20.84	1.11	28.23	1.89	11.88	2.60
8	TGT8	OBJ5	84.62	84.62	84.62	100.00	6.67	59.50	3.17	70.13	4.69	8.86	1.94
	TOTALS		385.00 48.12	385.00	48.12	659.28	43.95	844.95	45.06	677.24	45.30	220.30	48.31

WEAPON RESULTS

	WEAPON	TYPE	WEAPONS USED NUMBER	WEAPONS USED PERCENT	WEAPONS REMAINING NUMBER	WEAPONS REMAINING PERCENT
1	WPN1	ICBM	46.91	31.27	103.09	68.73
2	WPN2	ICBM	76.55	51.03	73.45	48.97
3	WPN3	ICBM	76.55	51.03	73.45	48.97
4	WPN4	ICBM	76.55	51.03	73.45	48.97
5	WPN5	SLBM	76.55	51.03	73.45	48.97
6	WPN6	SLBM	76.55	51.03	73.45	48.97
7	WPN7	SLBM	76.55	51.03	73.45	48.97
8	WPN8	BMBR	76.55	51.03	73.45	48.97
9	WPN9	BMBR	0.00	0.00	150.00	100.00
10	WPN10	BMBR	76.55	51.03	73.45	48.97
	TOTALS		659.28	43.95	840.72	56.05

LAYDOWN	NUMBER	DE	WEAPONS 1	2	3	4	5	6	7	8	9	10
TARGET 1	TGT1	OBJ1										
	76.55	.7460	0	0	0	0	0	0	1	0	0	0
	17.19	.7500	0	0	0	0	0	0	0	0	0	1
	6.26	0.0000	0	0	0	0	0	0	0	0	0	0
TOTALS=	100.00	.7000	0	0	0	0	0	0	77	0	0	17
TARGET 2	TGT2	OBJ1										
	100.00	0.0000	0	0	0	0	0	0	0	0	0	0
TOTALS=	100.00	0.0000	0	0	0	0	0	0	0	0	0	0
TARGET 3	TGT3	OBJ2										
	100.00	0.0000	0	0	0	0	0	0	0	0	0	0
TOTALS=	100.00	0.0000	0	0	0	0	0	0	0	0	0	0
TARGET 4	TGT4	OBJ3										
	76.55	.5787	0	0	0	1	0	0	0	0	0	0
	10.30	.2273	0	0	1	0	0	0	0	0	0	0
	10.98	.2849	0	0	1	0	0	1	0	0	0	0
	2.17	.4137	0	0	1	0	1	0	0	0	0	0
TOTALS=	100.00	.5000	0	0	23	77	2	11	0	0	0	0
TARGET 5	TGT5	OBJ3										
	23.53	0.0000	0	0	0	0	0	0	0	0	0	0
	76.47	.8500	0	1	0	0	0	0	0	0	0	0
TOTALS=	100.00	.6500	0	76	0	0	0	0	0	0	0	0
TARGET 6	TGT6	OBJ3										
	27.47	.6970	0	0	1	0	1	0	0	0	0	0
	46.91	.5949	1	0	0	0	1	0	0	0	0	0
	25.62	.5106	0	0	1	0	0	1	0	0	0	0
TOTALS=	100.00	.6000	47	0	53	0	74	26	0	0	0	0
TARGET 7	TGT7	OBJ4										
	59.36	.7277	0	0	0	0	0	0	0	0	0	1
	16.56	.7397	0	0	0	0	0	1	0	0	0	0
	24.08	0.0000	0	0	0	0	0	0	0	0	0	0
TOTALS=	100.00	.5538	0	0	0	0	0	17	0	0	0	59
TARGET 8	TGT8	OBJ5										
	.07	.8500	0	1	0	0	0	0	0	0	0	0
	23.38	.8300	0	0	0	0	0	1	0	0	0	0
	76.55	.8450	0	0	0	0	0	0	0	1	0	0
TOTALS=	100.00	.8462	0	0	0	0	0	23	0	77	0	0

SAMPLE CASE 2

The second sample case, also an RFS computation, uses the same weapon base and target base as the first sample case. However, it allows a maximum of two weapons per target (instead of one), except that attacks on OBJ3 targets by one SLBM weapon and one ICBM weapon are prohibited.

WEAPON ALLOCATION PROBLEM

ARSENAL

	NAME	TYPE	NUMBER	HOB	YIELD	CEP	PLS	WSR	PTP	EMT	CMP	N MT	N EMT	N CMP
1	WPN1	ICBM	150	A	3.000	4000	1.000	.950	1.000	1.732	.130	450.000	259.808	19.501
2	WPN2	ICBM	150	A	2.500	3000	1.000	.850	1.000	1.581	.205	375.000	237.171	30.700
3	WPN3	ICBM	150	A	2.000	2000	1.000	.750	1.000	1.414	.397	300.000	212.132	59.528
4	WPN4	ICBM	150	A	1.500	1000	1.000	.850	1.000	1.225	1.310	225.000	183.712	196.556
5	WPN5	SLBM	150	G	1.000	1500	1.000	.950	1.000	1.000	.444	150.000	150.000	66.667
6	WPN6	SLBM	150	G	.900	2500	1.000	.850	1.000	.932	.149	135.000	139.825	22.372
7	WPN7	SLBM	150	G	.700	3500	1.000	.750	1.000	.788	.064	105.000	118.256	9.654
8	WPN8	BMBR	150	G	.500	3000	1.000	.850	1.000	.630	.070	75.000	94.494	10.499
9	WPN9	BMBR	150	A	.300	2000	1.000	.950	1.000	.448	.112	45.000	67.221	16.805
10	WPN10	BMBR	150	A	.100	1000	1.000	.750	1.000	.215	.158	15.000	32.317	23.773
	TOTALS =		1500									1875.000	1494.935	456.054

TARGET SYSTEM

	NAME	TYPE	NUMBER	VNTK	VALUE	N VALUE	MIN DE	MAX DE	AVE DE	RADIUS
1	TGT1	OBJ1	100	10Q5	1.00	100.00	0.0000	.8000	.7000	0.0000
2	TGT2	OBJ1	100	12Q6	1.00	100.00	0.0000	.8000	0.0000	0.0000
3	TGT3	OBJ2	100	8P4	1.00	100.00	0.0000	.8000	0.0000	0.0000
4	TGT4	OBJ3	100	45P2	1.00	100.00	0.0000	.9000	.5000	0.0000
5	TGT5	OBJ3	100	16Q8	1.00	100.00	0.0000	.9000	.6500	0.0000
6	TGT6	OBJ3	100	40P5	1.00	100.00	0.0000	.9000	.6000	0.0000
7	TGT7	OBJ4	100	20Q2	1.00	100.00	0.0000	.8000	-1.0000	0.0000
8	TGT8	OBJ5	100	11P1	1.00	100.00	0.0000	.9000	.7000	0.0000
	TOTALS =		800			800.60				

UF TABLE

TARGET TYPE	WEAPON TYPE 1	2	3	4	5	6	7	8	9	10
1	.950000	.850000	.750000	.850000	.950000	.849990	.746035	.840637	.950000	.750000
2	.949998	.850000	.750000	.850000	.950000	.849706	.736094	.836949	.949926	.750000
3	.950000	.850000	.750000	.850000	.950000	.850000	.749992	.849996	.950000	.750000
4	.102285	.134463	.222273	.570680	.246089	.080581	.034006	.041096	.086545	.122389
5	.948766	.849960	.750000	.850000	.949998	.842779	.691034	.788883	.942241	.749401
6	.215404	.263584	.403513	.757404	.483693	.179602	.071907	.086612	.172774	.227866
7	.872877	.824481	.747531	.850000	.938613	.735664	.483464	.568793	.847318	.727743
8	.950000	.850000	.750000	.850000	.950000	.849980	.743742	.845039	.950000	.750000

THE SOLUTION IS LIMITED TO THE FOLLOWING WEAPON VECTORS

ALL POSSIBLE WEAPON VECTORS WITH 2 OR LESS WEAPONS

MINUS ALL VECTORS OF THE FOLLOWING TYPES

FOR THE FOLLOWING TARGETS ONLY OBJ3

1 ALL POSSIBLE WEAPON VECTORS WITH

1 SLBM WEAPONS

AND 1 ICBM WEAPONS

	C	A MATRIX 1	2	3	4	5	6	7	8	9	1A	1	2	3	4	5	6	7	8					
1	0.00	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
2	0.00	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
3	0.00	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
4	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
5	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
7	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
8	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
9	0.00	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
10	0.00	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
11	0.00	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
12	0.00	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
13	0.00	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
14	0.00	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
15	0.00	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
16	0.00	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
17	0.00	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
18	0.00	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
19	0.00	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
20	0.00	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
21	0.00	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
22	0.00	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
23	0.00	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
24	0.00	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
25	0.00	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
26	0.00	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
27	0.00	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
28	0.00	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
29	0.00	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
30	0.00	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
31	0.00	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
32	0.00	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
33	0.00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
34	0.00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
35	0.00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
36	0.00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
37	0.00	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
38	0.00	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
39	0.00	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
40	0.00	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
41	0.00	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
42	0.00	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
43	0.00	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
44	0.00	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
45	0.00	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
46	0.00	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
47	0.00	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
48	0.00	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
49	0.00	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
50	0.00	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
51	0.00	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
52	0.00	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
53	0.00	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
54	0.00	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
55	0.00	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
56	0.00	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

	C	A MATRIX																						
		1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8					
57	0.00	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0
58	0.00	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0
59	0.00	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0
60	0.00	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0
61	0.00	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
62	0.00	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
63	0.00	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
64	0.00	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
65	0.00	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
66	0.00	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
67	0.00	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
68	0.00	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
69	0.00	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
70	0.00	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
71	0.00	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
72	0.00	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
73	0.00	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
74	0.00	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
75	0.00	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
76	0.00	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
77	0.00	0	0	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0
78	0.00	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0
79	0.00	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
80	0.00	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
81	0.00	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
82	0.00	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
83	0.00	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
84	0.00	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
85	0.00	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
86	0.00	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
87	0.00	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
88	0.00	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
89	0.00	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
90	0.00	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
91	0.00	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
92	0.00	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
93	0.00	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
94	0.00	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
95	0.00	0	0	0	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
96	0.00	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
97	0.00	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
98	0.00	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
99	0.00	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
100	0.00	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
101	0.00	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
102	0.00	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
103	0.00	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
104	0.00	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
105	0.00	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
106	0.00	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
107	0.00	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
108	0.00	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
109	0.00	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
110	0.00	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
111	0.00	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
112	0.00	0	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

C A MATRIX

	C	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8					
113	0.00	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
114	0.00	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
115	0.00	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
116	0.00	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
117	0.00	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
118	0.00	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
119	0.00	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
120	0.00	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
121	0.00	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
122	0.00	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
123	0.00	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
124	0.00	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
125	0.00	0	0	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
126	0.00	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
127	0.00	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
128	0.00	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
129	0.00	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
130	0.00	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
131	0.00	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
132	0.00	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
133	0.00	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
134	0.00	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
135	0.00	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
136	0.00	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
137	0.00	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
138	0.00	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
139	0.00	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
140	0.00	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
141	0.00	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
142	0.00	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
143	0.00	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
144	0.00	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
145	0.00	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
146	0.00	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
147	0.00	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
148	0.00	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
149	0.00	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
150	0.00	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
151	0.00	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
152	0.00	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
153	0.00	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
154	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
155	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
156	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
157	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
158	0.00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	B =																							
		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1
		5	5	5	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	7	5	6	0	4
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0

1	7	24.211396
2	136	14.918114
3	135	13.595649
4	35	25.011653
5	92	58.349652
6	24	71.945301
7	36	28.054699
8	130	30.754016
9	23	71.945301
10	9	21.768351
11	159	.479635
12	2	100.000000
13	3	100.000000
14	87	14.513438
15	33	18.278949
16	86	57.431863
17	15	50.176950
18	142	13.136585
19	1	6.286348
20	133	21.715769
21	5	23.371399
22	124	7.256719
23	39	26.677846
24	160	.479635

THE RELATIVE FORCE SIZE IS 2.00

UNCLASSIFIED

PROBLEM 2 OPTIMAL SOLUTION FOR MAXIMUM VALUE DAMAGED

TARGET RESULTS

	TARGET	TYPE	VALUE DAMAGED	TARGETS DAMAGED NUMBER	TARGETS DAMAGED PERCENT	WEAPONS USED NUMBER	WEAPONS USED PERCENT	MEGATONS USED NUMBER	MEGATONS USED PERCENT	EMT USED NUMBER	EMT USED PERCENT	CMP USED NUMBER	CMP USED PERCENT
1	TGT1	OBJ1	70.00	70.00	70.00	93.71	6.25	52.54	2.80	61.41	4.11	8.08	1.77
2	TGT2	OBJ1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	TGT3	OBJ2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	TGT4	OBJ3	50.00	50.00	50.00	173.32	11.55	357.94	19.09	240.33	16.08	111.17	24.38
5	TGT5	OBJ3	65.00	65.00	65.00	76.63	5.11	162.33	8.66	109.30	7.31	14.07	3.22
6	TGT6	OBJ3	60.00	60.00	60.00	200.00	13.33	237.22	12.65	199.83	13.37	63.02	13.99
7	TGT7	OBJ4	55.36	55.36	55.36	75.79	5.05	28.07	1.50	34.68	2.32	11.77	2.58
8	TGT8	OBJ5	84.64	84.64	84.64	100.00	6.67	61.22	3.27	71.47	4.78	9.22	2.02
	TOTALS		385.00 48.12	385.00	48.12	719.45	47.96	899.32	47.96	717.02	47.96	218.74	47.96

WEAPON RESULTS

	WEAPON	TYPE	WEAPONS USED NUMBER	WEAPONS USED PERCENT	WEAPONS REMAINING NUMBER	WEAPONS REMAINING PERCENT
1	WPN1	ICBM	71.95	47.96	78.05	52.04
2	WPN2	ICBM	71.95	47.96	78.05	52.04
3	WPN3	ICBM	71.95	47.96	78.05	52.04
4	WPN4	ICBM	71.95	47.96	78.05	52.04
5	WPN5	SLBM	71.95	47.96	78.05	52.04
6	WPN6	SLBM	71.95	47.96	78.05	52.04
7	WPN7	SLBM	71.95	47.96	78.05	52.04
8	WPN8	BMBR	71.95	47.96	78.05	52.04
9	WPN9	BMBR	71.95	47.96	78.05	52.04
10	WPN10	BMBR	71.95	47.96	78.05	52.04
	TOTALS		719.45	47.96	780.55	52.04

UNCLASSIFIED

LAYDOWN NUMBER	DE	WEAPONS 1	2	3	4	5	6	7	8	9	10
TARGET 1 TGT1	UAJ1										
71.95	.7460	0	0	0	0	0	0	1	0	0	0
21.77	.7500	0	0	0	0	0	0	0	0	0	1
6.29	0.0000	0	0	0	0	0	0	0	0	0	0
TOTALS= 100.00	.7000	0	0	0	0	0	0	72	0	0	22
TARGET 2 TGT2	UAJ1										
100.00	0.0000	0	0	0	0	0	0	0	0	0	0
TOTALS= 100.00	0.0000	0	0	0	0	0	0	0	0	0	0
TARGET 3 TGT3	UAJ2										
100.00	0.0000	0	0	0	0	0	0	0	0	0	0
TOTALS= 100.00	0.0000	0	0	0	0	0	0	0	0	0	0
TARGET 4 TGT4	UAJ3										
14.92	.3018	1	0	1	0	0	0	0	0	0	0
30.75	.6146	1	0	0	1	0	0	0	0	0	0
14.51	.6078	0	0	0	1	0	0	0	0	1	0
13.14	.1941	2	0	0	0	0	0	0	0	0	0
26.68	.5767	0	0	0	1	0	0	0	0	0	0
TOTALS= 100.00	.5000	72	0	15	72	0	0	0	0	15	0
TARGET 5 TGT5	UAJ3										
58.35	.8500	0	1	0	0	0	0	0	0	0	0
18.28	.8028	0	0	0	0	0	1	0	0	0	0
23.37	0.0000	0	0	0	0	0	0	0	0	0	0
TOTALS= 100.00	.6500	0	58	0	0	0	18	0	0	0	0
TARGET 6 TGT6	UAJ3										
13.60	.5727	0	1	1	0	0	0	0	0	0	0
57.43	.5729	0	0	0	0	1	0	0	0	1	0
21.72	.6442	0	0	2	0	0	0	0	0	0	0
7.26	.7434	0	0	0	0	2	0	0	0	0	0
TOTALS= 100.00	.6000	0	14	57	0	72	0	0	0	57	0
TARGET 7 TGT7	UAJ4										
24.21	0.0000	0	0	0	0	0	0	0	0	0	0
25.61	.7357	0	0	0	0	0	1	0	0	0	0
50.18	.7677	0	0	0	0	0	0	0	0	0	1
TOTALS= 100.00	.5536	0	0	0	0	0	26	0	0	0	50
TARGET 8 TGT8	UAJ5										
29.05	.8500	0	0	0	0	0	1	0	0	0	0
71.95	.8480	0	0	0	0	0	0	0	1	0	0
TOTALS= 100.00	.8464	0	0	0	0	0	28	0	72	0	0

SAMPLE CASE 3

The third sample case maximizes target value destroyed. It allows a maximum of two weapons per target, and incorporates the collateral damage feature wherein a given weapon is assigned to more than one target.

WEAPON ALLOCATION PROBLEM 3

ARSENAL

	NAME	TYPE	NUMBER	MOR	YIELD	CEP	PLS	WSR	PTP	EMT	CMP	N MT	N EMT	N CMP
1	WPN1		700	A	5.000	3000	.900	.500	1.000	2.236	.325	3500.000	1565.248	227.424
2	WPN2		500	A	3.000	2500	.900	.600	.900	1.732	.333	1500.000	866.025	166.407
3	WPN3		300	A	1.000	2000	.900	.700	.800	1.000	.250	300.000	300.000	75.000
4	WPN4		500	A	.500	1500	.900	.600	.700	.630	.280	250.000	314.980	139.991
5	WPN5		700	A	.200	1000	.900	.500	.600	.342	.342	140.000	239.397	239.397
	TOTALS =		2700									5690.000	3285.650	848.218

TARGET SYSTEM

	NAME	TYPE	NUMBER	VNTK	VALUE	N VALUE	MIN DE	MAX DE	AVE DE	RADIUS
1	TGT1		10	1005	1.00	10.00	0.0000	1.0000	0.0000	0.0000
2	TGT2		20	1005	.50	10.00	0.0000	1.0000	0.0000	0.0000
3	TGT3		25	1005	.40	10.00	0.0000	1.0000	0.0000	0.0000
4	TGT4		40	1005	.25	10.00	0.0000	1.0000	0.0000	0.0000
5	TGT5		50	1005	.20	10.00	0.0000	1.0000	0.0000	0.0000
6	TGT6		100	1005	.10	10.00	0.0000	1.0000	0.0000	0.0000
7	TGT7		200	1005	.05	10.00	0.0000	1.0000	0.0000	0.0000
8	TGT8		250	1005	.04	10.00	0.0000	1.0000	0.0000	0.0000
9	TGT9		500	1005	.02	10.00	0.0000	1.0000	0.0000	0.0000
10	TGT10		1000	1005	.01	10.00	0.0000	1.0000	0.0000	0.0000
	TOTALS =		2195			100.00				

UF TABLE

TARGET TYPE	WEAPON TYPE 1	2	3	4	5
1	.450000	.480000	.504000	.378000	.270000
2	.450000	.480000	.504000	.378000	.270000
3	.450000	.480000	.504000	.378000	.270000
4	.450000	.480000	.504000	.378000	.270000
5	.450000	.480000	.504000	.378000	.270000
6	.450000	.480000	.504000	.378000	.270000
7	.450000	.480000	.504000	.378000	.270000
8	.450000	.480000	.504000	.378000	.270000
9	.450000	.480000	.504000	.378000	.270000
10	.450000	.480000	.504000	.378000	.270000

THE SOLUTION IS LIMITED TO THE FOLLOWING WEAPON VECTORS

ALL POSSIBLE WEAPON VECTORS WITH 2 OR LESS WEAPONS

PROBLEM 3 OPTIMAL SOLUTION FOR MAXIMUM VALUE DAMAGED

TARGET RESULTS

	TARGET	TYPE	VALUE DAMAGED	TARGETS DAMAGED NUMBER	TARGETS DAMAGED PERCENT	WEAPONS USED NUMBER	WEAPONS USED PERCENT	MEGATONS USED NUMBER	MEGATONS USED PERCENT	EMT USED NUMBER	EMT USED PERCENT	CMP USED NUMBER	CMP USED PERCENT
1	TGT1		7.54	7.54	15.42	6.85	.25	6.85	.12	6.85	.21	1.71	.20
2	TGT2		7.54	15.08	15.40	13.69	.51	13.69	.24	13.69	.42	3.42	.40
3	TGT3		7.54	18.85	15.42	17.12	.63	17.12	.30	17.12	.52	4.28	.50
4	TGT4		7.54	30.16	15.42	27.39	1.01	27.39	.48	27.39	.83	6.85	.81
5	TGT5		7.54	37.70	15.42	34.24	1.27	34.24	.60	34.24	1.04	8.56	1.01
6	TGT6		7.54	75.40	15.42	68.47	2.54	68.47	1.20	68.47	2.08	17.12	2.02
7	TGT7		7.46	149.14	14.57	114.25	4.23	218.93	3.85	152.56	4.64	32.90	3.88
8	TGT8		7.45	186.26	14.51	140.68	5.21	281.35	4.94	192.17	5.85	40.99	4.83
9	TGT9		7.36	367.90	13.50	237.47	8.80	712.40	12.52	411.31	12.52	79.03	9.32
10	TGT10		7.10	710.60	11.02	423.36	15.68	1837.10	32.29	876.18	26.67	138.65	16.35
	TOTALS		74.61	1598.23	12.81	1083.51	40.13	3217.54	56.55	1799.97	54.78	333.52	39.32
			74.61										

WEAPON RESULTS

	WEAPON	TYPE	WEAPONS USED NUMBER	WEAPONS USED PERCENT	WEAPONS REMAINING NUMBER	WEAPONS REMAINING PERCENT
1	WPN1		283.51	40.50	416.49	59.50
2	WPN2		600.00	100.00	.00	.00
3	WPN3		300.00	100.00	.00	.00
4	WPN4		0.00	0.00	500.00	100.00
5	WPN5		0.00	0.00	700.00	100.00
	TOTALS		1083.51	40.13	1616.49	59.87

LAYDOWN NUMBER		DE	WEAPONS 1	2	3	4	5
TARGET 1 TGT1							
	3.42	2.2023	0	0	2	0	0
TOTALS=	10.00	.7540	0	0	7	0	0
TARGET 2 TGT2							
	6.85	2.2023	0	0	2	0	0
TOTALS=	20.00	.7540	0	0	14	0	0
TARGET 3 TGT3							
	8.56	2.2023	0	0	2	0	0
TOTALS=	25.00	.7540	0	0	17	0	0
TARGET 4 TGT4							
	13.69	2.2023	0	0	2	0	0
TOTALS=	40.00	.7540	0	0	27	0	0
TARGET 5 TGT5							
	17.12	2.2023	0	0	2	0	0
TOTALS=	50.00	.7540	0	0	34	0	0
TARGET 6 TGT6							
	34.24	2.2023	0	0	2	0	0
TOTALS=	100.00	.7540	0	0	68	0	0
TARGET 7 TGT7							
	4.78	2.2023	0	0	2	0	0
	52.34	2.6481	0	1	1	0	0
TOTALS=	400.00	.7457	0	52	62	0	0
TARGET 8 TGT8							
	70.34	2.6481	0	1	1	0	0
TOTALS=	250.00	.7451	0	70	70	0	0
TARGET 9 TGT9							
	118.73	3.0985	0	2	0	0	0
TOTALS=	500.00	.7358	0	237	0	0	0
TARGET 10 TGT10							
	71.83	3.4819	2	0	0	0	0
	139.85	3.2899	1	1	0	0	0
TOTALS=	1000.00	.7182	284	140	0	0	0

APPENDIX A

MATHEMATICAL THEORY

Appendix A

MATHEMATICAL THEORY

The mathematical theory employed in ALLOT is best illustrated by an example. Suppose there are two target classes, consisting of T_1 and T_2 targets, respectively. Suppose further that there are W identical weapons, each with a single-shot probability of kill (SSPK) of K_1 against targets in the first target class, and K_2 against targets in the second class. Let V_1 and V_2 be the values of targets in the first and second target classes, respectively, and assume that it is not permitted to place more than two weapons against any individual target. With these constraints, consider the following two questions:

(1) What is the maximum possible value destroyed?

(2) If it is desired to achieve average damage levels of at least D_1 and D_2 against the two target classes, respectively, what is the relative force size (RFS)?

Each of these two examples is amenable to solution by first expressing the problem in a general linear programming form, and then solving it by standard linear programming techniques.[1]

---

[1]In general, a noninteger solution will be obtained, although the answers may be truncated or rounded off by the user. In view of the aggregation usually involved in the preparation of the target data base, such activity is not likely in practice to have a significant effect on the accuracy of the solution.

The general linear programming form used in ALLOT is as follows:

Find values of the variables $x_i (i=1,2,3,\ldots,m)$ and $y_j (j=1,2,3,\ldots,n)$ such that, for given a_{ij} and b_i

$$\sum_{j=1}^{n} a_{ij}y_j + x_i = b_i \quad (i=1,2,3,\ldots,m) \tag{1}$$

$$x_i \geq 0, \; y_j \geq 0 \tag{2}$$

$$x_1 = \text{minimum} \tag{3}$$

The algorithm used to solve these equations is described in Appendix B. The purpose of the present appendix is to show that each of the two examples presented above can be expressed in the general form of equations (1)-(3).

Consider the first problem - i.e., what is the maximum possible value destroyed? Define the following:

x_1 = the total value surviving (i.e., not destroyed)

x_2, y_1, y_2 = the numbers of targets in target class 1 assigned 0, 1, and 2 weapons, respectively.

x_3, y_3, y_4 = the numbers of targets in target class 2 assigned 0, 1, and 2 weapons, respectively.

x_4 = the number of weapons not used.

Then the requirements that the total numbers of targets in each class are T_1 and T_2, respectively, can be written

$$y_1 + y_2 + x_2 = T_1 \tag{4}$$

$$y_3 + y_4 + x_3 = T_2 \tag{5}$$

The requirement that the total number of weapons equals W is expressed as

$$y_1 + 2y_2 + y_3 + 2y_4 + x_4 = W \tag{6}$$

The total value surviving, x_1, is given by

$$x_1 = V_1[x_2 + y_1(1-K_1) + y_2(1-K_1)^2] + V_2[x_3 + y_3(1-K_2) + y_4(1-K_2)^2] \tag{7}$$

where V_1 and V_2 are the individual target values, and K_1 and K_2 are the SSPKs. If the expressions given in equations (4) and (5) for x_2 and x_3 are substituted into equation (7), the result is

$$x_1 = V_1[(T_1 - y_1 - y_2) + y_1(1-K_1) + y_2(1-K_1)^2] + V_2[(T_2 - y_3 - y_4) + y_3(1-K_2) + y_4(1-K_2)^2] \tag{8}$$

Rearranging terms in equation (8) gives

$$V_1[1-(1-K_1)]y_1 + V_1[1-(1-K_1)^2]y_2 + V_2[1-(1-K_2)]y_3 + V_2[1-(1-K_2)^2]y_4 + x_1 = V_1T_1 + V_2T_2 \tag{9}$$

Equations (9), (4), (5), and (6) comprise the desired set of conditions. Comparison with equations (1)-(3) shows that

$$m = 4 \qquad a_{22} = 1$$
$$n = 4 \qquad a_{23} = 0$$
$$b_1 = V_1T_1 + V_2T_2 \qquad a_{24} = 0$$
$$b_2 = T_1 \qquad a_{31} = 0$$
$$b_3 = T_2 \qquad a_{32} = 0$$
$$b_4 = W \qquad a_{33} = 1$$
$$a_{11} = V_1[1-(1-K_1)] \qquad a_{34} = 1$$
$$a_{12} = V_1[1-(1-K_1)^2] \qquad a_{41} = 1$$
$$a_{13} = V_2[1-(1-K_2)] \qquad a_{42} = 2$$
$$a_{14} = V_2[1-(1-K_2)^2] \qquad a_{43} = 1$$
$$a_{21} = 1 \qquad a_{44} = 2$$

This completes the discussion of the first problem. The technique shown is quite general and is easily extended to accommodate cases involving additional target classes and/or weapon classes.

Consider now the second problem - i.e., what is the RFS? In this case, the procedure followed by ALLOT is to first determine the forces required to meet the specified goals, and then determine the ratio of these forces to those available.

Use the same definitions as before, except for x_1. Redefine x_1 as the inverse of the RFS:

$$x_1 = 1/RFS$$

$x_2, x_3, x_4, y_1, y_2, y_3, y_4$ = same definitions as in first problem

Define, in addition,

y_5 = 1/RFS

x_5 = average damage in excess of required average damage D_1 to targets in target class 1

x_6 = average damage in excess of required average damage D_2 to targets in target class 2

Equations (4) and (5) are still applicable and are therefore retained. In equation (6), it is necessary to replace W, the number of weapons available, by Wy_5, the number of weapons actually required:

$$y_1 + 2y_2 + y_3 + 2y_4 + x_4 = Wy_5 \tag{10}$$

The equations describing x_5 and x_6 deal with the total value surviving in the two target classes, respectively, and are somewhat similar in form to equation (7):

$$V_1T_1(1-D_1-x_5) = V_1[x_2 + y_1(1-K_1) + y_2(1-K_1)^2] \tag{11}$$

$$V_2T_2(1-D_2-x_6) = V_2[x_3 + y_3(1-K_2) + y_4(1-K_2)^2] \tag{12}$$

With substitutions from equations (4) and (5), and rearrangement of terms, equations (11) and (12) become

$$\frac{[-1+(1-K_1)]y_1 + [-1+(1-K_1)^2]y_2}{T_1} + x_5 = -D_1 \tag{13}$$

$$\frac{[-1+(1-K_2)]y_3 + [-1+(1-K_2)^2]y_4}{T_2} + x_6 = -D_2 \tag{14}$$

Finally, the definitions of x_1 and y_5 require that

$$-y_5 + x_1 = 0 \tag{15}$$

Accordingly, equations (15), (4), (5), (10), (13), and (14) comprise the desired set of conditions. Comparison with equations (1)-(3) shows that

$m = 6$	$a_{14} = 0$	$a_{34} = 1$	$a_{55} = 0$
$n = 5$	$a_{15} = 1$	$a_{35} = 0$	$a_{61} = 0$
$b_1 = 0$	$a_{21} = 1$	$a_{41} = 1$	$a_{62} = 0$
$b_2 = T_1$	$a_{22} = 1$	$a_{42} = 2$	$a_{63} = [-1+(1-K_2)]/T_2$
$b_3 = T_2$	$a_{23} = 0$	$a_{43} = 1$	$a_{64} = [-1+(1-K_2)^2]/T_2$
$b_4 = 0$	$a_{24} = 0$	$a_{44} = 2$	$a_{65} = 0$
$b_5 = D_1$	$a_{25} = 0$	$a_{45} = -W$	
$b_6 = D_2$	$a_{31} = 0$	$a_{51} = [-1+(1-K_1)]/T_1$	
$a_{11} = 0$	$a_{32} = 0$	$a_{52} = [-1+(1-K_1)^2]/T_1$	
$a_{12} = 0$	$a_{33} = 1$	$a_{53} = 0$	
$a_{13} = 0$		$a_{54} = 0$	

This completes the discussion of the second problem. Once again, the technique is quite general and is easily extended to accommodate cases involving additional target classes and/or weapon classes.

The algorithm used to solve the general set of equations (1)-(3) is presented in Appendix B.

APPENDIX B

ALLOT LINEAR PROGRAMMING ALGORITHM

Appendix B

ALLOT LINEAR PROGRAMMING ALGORITHM

Consider the following linear programming problem: Find values of the variables x_i $(i = 1,2,3,\ldots,m)$ and $y_j (j = 1,2,3,\ldots n)$ such that, for given a_{ij} and b_i,

$$\sum_{j=1}^{n} a_{ij}y_j + x_i = b_i \tag{1}$$

$$x_i \geq 0,\ y_j \geq 0 \tag{2}$$

$$x_1 = \text{minimum} \tag{3}$$

(A trial solution is

$$x_i = b_i,\ y_j = 0$$

This is, in fact, the desired answer if $b_i \geq 0$ for all i, and if $a_{1j} \leq 0$ for all j. If these conditions are not met, however, further work is necessary.)

Given equations (1)-(3), the steps of the algorithm used in ALLOT[1] to solve this problem are as follows:

1. If $b_i \geq 0$ for all $i \geq 2$, go to step 7.
2. Find an $i \geq 2$ for which $b_i \leq 0$. Call it α.
3. Find the value of j for which $a_{\alpha j}$ is the smallest. Call it β. Then $a_{\alpha\beta} \leq a_{\alpha j}$ for all j.
4. If $a_{\alpha\beta} \geq 0$, the problem as stated above in equations (1) - (3) has no solution.

[1]This algorithm is a variant of the well-known simplex algorithm.

5. Consider all $i \geq 2$ for which $b_i \geq 0$, and $a_{i\beta} > 0$. Of these, find the value of i for which $(b_i/a_{i\beta})$ is the smallest. Call it γ. (If there is no $i \geq 2$ such that $b_i \geq 0$ and $a_{i\beta} > 0$, set γ equal to α.)
6. Go to step 10.
7. (Via step 1). If $a_{1j} \leq 0$ for all j, the problem is solved. The solution is $x_i = b_i$, $y_j = 0$.
8. If there exists an $a_{1j} > 0$, then find the value of j for which a_{1j} is the largest. Call it β. Then $a_{1\beta} \geq a_{1j}$ for all j, and $a_{1\beta} > 0$.
9. Consider all $i \geq 2$ for which $a_{i\beta} > 0$. Of these, find the value of i for which $(b_i/a_{i\beta})$ is the smallest. Call it γ. (If there is no $i \geq 2$ for which $a_{i\beta} > 0$, the problem as stated above in equations (1)-(3) is unbounded.)
10. (Via step 6 or step 9). Consider equation (1) with $i = \gamma$:

$$\sum_j a_{\gamma j} y_j + x_\gamma = b_\gamma \tag{4}$$

11. Divide both sides of this equation by $a_{\gamma\beta}$:

$$\sum_{j \neq \beta} (a_{\gamma j}/a_{\gamma\beta}) y_j + (1/a_{\gamma\beta}) x_\gamma + y_\beta = (b_\gamma/a_{\gamma\beta}) \tag{5}$$

$$\text{i.e., } y_\beta = (b_\gamma/a_{\gamma\beta}) - \sum_{j \neq \beta} (a_{\gamma j}/a_{\gamma\beta}) y_j - (1/a_{\gamma\beta}) x_\gamma \tag{6}$$

12. Use equation (6) to substitute for y_β in the remainder of equations (1):

$$\sum_{j \neq \beta} [a_{ij} - (a_{i\beta} a_{\gamma j}/a_{\gamma\beta})] y_j - (a_{i\beta}/a_{\gamma\beta}) x_\gamma + x_i =$$

$$b_i - (a_{i\beta} b_\gamma/a_{\gamma\beta}) \quad (i \neq \gamma) \tag{7}$$

13. Define $u_i \equiv x_i$ for all $i \neq \gamma$; $u_\gamma \equiv y_\beta$; $v_j \equiv y_j$ for all $j \neq \beta$; and $v_\beta \equiv x_\gamma$.

14. Define $e_{ij} \equiv a_{ij} - (a_{i\beta} a_{\gamma j}/a_{\gamma\beta})$ for all $i \neq \gamma$ and $j \neq \beta$; $e_{i\beta} \equiv -(a_{i\beta}/a_{\gamma\beta})$ for all $i \neq \gamma$; $e_{\gamma j} \equiv (a_{\gamma j}/a_{\gamma\beta})$ for all $j \neq \beta$; and $e_{\gamma\beta} = (1/a_{\gamma\beta})$.

15. Define $d_i \equiv b_i - (a_{i\beta} b_\gamma/a_{\gamma\beta})$ for all $i \neq \gamma$; and $d_\gamma \equiv b_\gamma/a_{\gamma\beta}$).

16. With the above definitions equations (1)-(3) may be written via equations (5)-(7) as:

$$\sum_j e_{ij} v_j + u_i = d_i \qquad (9)$$

$$u_i \geq 0, \; v_j \geq 0 \qquad (10)$$

$$u_1 = \text{Minimum} \qquad (11)$$

(Equation (11) follows because $\gamma \geq 2$, and therefore $u_1 = x_1$.)

17. Equations (9)-(11) have exactly the same form as equations (1)-(3). Thus, the above procedure--i.e., steps (1)-(16)--may be applied again. The next step, therefore, is to go back to step 1 and iterate (using the e_{ij}, v_j, u_i, and d_i, respectively, instead of the a_{ij}, y_j, x_i, and b_i).

The above is the algorithm used in ALLOT to solve the allocation problem. The detailed coding uses a matrix framework to save computer storage, but in all other respects is identical to the description given here.

APPENDIX C

LISTING OF ALLOT PROGRAM

```
      PROGRAM ALLOT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      COMMON /BLKA/ B(51),C(20U1),M,N,NZ
      COMMON /BLKB/ NP,NT,NW,NV,IOBJ
      COMMON /BLKC/ JA(51),XA(51)
      COMMON /BKLD/ PK(20,20),FID(20,20)
      COMMON /BLKE/ PUN,DSC,DIN,MAT,VEC,NOL,NUS,TITLE(10),OBJCON(5),IDSR
      COMMON /BLKF/ N1,N2,N3,INSC(10),INRL(6,10),INSW(6,10),
     *              INLR(5,10),INUB(5,10),INAP(20,10)
      COMMON /BLKG/ FSTENT,MATEUI,VECFND,FINISH,SKPCHK
      COMMON /BLKH/ NTSUM,NWSUM,VSUM,YLDSUM,ENTSUM,CMPSUM
      COMMON /BLKI/ V(21),DEMIN(21),DEMAX(21),DEAVE(21),DEDLT(21),
     *              YLD(21),ENT(21),CMP(21)
      COMMON /BLKJ/ WNAME(2,21),TNAME(2,21),WTYPE(21),TTYPE(21),
     *              NUMW(21),NUMT(21),VN(21),PU(21),AD(21),CEP(21),
     *              PLS(21),WSN(21),PTP(21),MUB(21)
      COMMON /BLKL/ IR(20),IAP(20)
      COMMON /BLKN/ NSENT,ISUM(2,20),XSUM(13,20)
      COMMON /BLKY/ JSRT(50),JEND(20)
      COMMON /BLKZ/ IA(4,2000),UA(2000),TA(2000),NK
      COMMON /RED / NVW,KA,F
      COMMON /BLUE/ Q(51,51),M1
      DIMENSION IOCONT(7)
      EQUIVALENCE (PUN,IOCONT(1))
      LOGICAL PUN,DSC,DIN,MAT,VEC,NOL,NOS,OBJCON,IOCONT
      NP=0
      IDSR=5
      DO 900 I=1,7
  900 IOCONT(I)=.FALSE.
 1000 CALL INPUT
 1010 DO 1020 I=1,5
      IF(OBJCON(I))GO TO 1030
 1020 CONTINUE
      GO TO 1000
 1030 IOBJ=I
      OBJCON(I)=.FALSE.
      NV=0
 1100 CALL BUILDA
 1500 IF(MAT)CALL OUTMAT
      CALL SAMPLE
      IF(MAT) CALL DUAL
 1700 CALL OUTPUT
 6000 GO TO 1010
      END
```

```
      SUBROUTINE INPUT
C
C NOTE -- DEOLT IS THE TARGET RADIUS, IN NAUTICAL MILES.
C
      COMMON /BLKA/ B(51),C(2001),M,N,NZ
      COMMON /BLKB/ NP,NT,NW,NV,IOB
      COMMON /BKLD/ PK(20,20),FIP(20,20)
      COMMON /BLKE/ PUN,DSC,DIN,MAT,VEC,NOL,NOS,TITLE(10),OBJCON(5),IDGR
      COMMON /BLKF/ N1,N2,N3,INSP(10),INBL(6,10),INSW(6,10),
     *              INLB(5,10),INUB(5,10),INAP(20,10)
      COMMON /BLKH/ NTSUM,NWSUM,VSUM,YLDSUM,EMTSUM,CMPSUM
      COMMON /BLKI/ V(21),DEMIN(21),DEMAX(21),DEAVE(21),DEOLT(21),
     *              YLD(21),EMT(21),CMP(21)
      COMMON /BLKJ/ WNAME(2,21),TNAME(2,21),WTYPE(21),TTYPE(21),
     *              NUMW(21),NUMT(21),VN(21),PU(21),AD(21),CEP(21),
     *              PLS(21),WSR(21),PTP(21),HOB(21)
      COMMON /BLKN/ NSENT,ISUM(2,20),XSUM(13,20)
      COMMON /BLKY/ JSRT(20),JEND(20)
      INTEGER HOB,VN,AD
      LOGICAL PUN,DSC,DIN,MAT,VEC,NOL,NOS,OBJCON,IOCONT,CDE
      DIMENSION IAB(6),IAP(20,2),ITA(40,2),NB(2),NC(2),IFLD(7),XN(2,6)
      DIMENSION IOCONT(7),LCON(9),LOBJ(5),LIO(7),LNUM(10)
      DIMENSION TABLE(2,40,2),ILB(40,2),IUB(40,2),NENT(2),NAS(2)
      DIMENSION XTABL(2,40)
      DIMENSION IX(10),JX(10),DUM(10)
      EQUIVALENCE (PUN,IOCONT(1)),(NWENT,NENT(1)),(NTENT,NENT(2)),
     *            (XBLANK,IBLANK),(N2,NAS(1))
      DATA IGREAT,ILESS,IFOR,INOUT/1HG,1HL,6H*FOR,6H*IN//
      DATA LNUM/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0/
      DATA ISTART,IBLANK,IAND,LG,LL/6H*STA,6H      ,6HAND ,1HG,1HL/
      DATA LCON/6H*WPN,6H*TGT,6H*NW,6H*NTG,6H*ADD,6H*SUB,6H*CWP,6H*CTG,
     *        6H*END/
      DATA LOBJ/3HDE ,3HWPN,3HMT ,3HEMT,3HCMP/
      DATA LIO/3HPUN,3HDSC,3HDIN,3HMAT,3HVEC,3HNOL,3HNOS/
      DATA XALL,XEXC,XOTA,XOTB/6HALL ,6HEXC ,6HOTHE,6HRS  /
      DATA XW,LAIR/1HW,1HA/
      DATA N2,N3,NSENT/0,0,0/
      DATA IFT/0/
      DATA EPS/0.000000001/
  910 FORMAT(20I4)
  914 FORMAT(A4,3X,A1,7(1X,A3),4X,10A4)
  915 FORMAT(1X,I2,2X,2A4,2X,A4,I8,3X,A1,F9.3,F8.0,3F8.3,2F10.3,3F12.3)
  916 FORMAT(1X,I2,2X,2A4,2X,A4,I8,I5,A1,I1,F9.2,F12.2,4F11.4)
  917 FORMAT(/5X,*TOTALS =*,I14,16X,F12.2)
  918 FORMAT(3X,I2,1X,10(2X,F8.6))
  921 FORMAT(*1*,40X,*     WEAPON ALLOCATION PROBLEM *,I3,*      */
     *        41X,10A4//)
  922 FORMAT(61X,*ARSENAL*//7X,*NAME     TYPE  NUMBER  HOB   YIELD*,4X,
     *  *CEP     PLS     WSR     PTP      EMT       CMP        N MT*,
     *        8X,*N EMT        N CMP*/)
  923 FORMAT(58X,*TARGET SYSTEM*//7X,*NAME      TYPE  NUMBER   VNTK*,
     *        4X,*VALUE      N VALUE      MIN DE       MAX DE      AVE DE*,
     *        5X,*RADIUS*/)
  924 FORMAT(*1*,21X,*DE TABLE*)
  925 FORMAT(/1X,*TARGET*,14X,*WEAPON TYPE*)
  926 FORMAT(2X,*TYPE*,5X,10(I2,8X)/)
  927 FORMAT(//1X,*THE SOLUTION IS LIMITED TO THE FOLLOWING WEAPON VECTO
```

```
     1RS */)
 928 FORMAT(4X,*  ALL POSSIBLE WEAPON VECTORS WITH *,I2,* OR LESS WEAPO
     1NS */)
 929 FORMAT(4X,*  PLUS ALL VECTORS OF THE FOLLOWING TYPES *)
 931 FORMAT(12X,A4,A1,I1,1X,2A4,* WEAPONS*)
 934 FORMAT(/4X,*  MINUS ALL VECTORS OF THE FOLLOWING TYPES *)
 937 FORMAT(/5X,*TOTALS =*,I14,40X,3F12.3///)
 938 FORMAT(/6X,*  FOR THE FOLLOWING TARGETS ONLY    *,6(2A4,*,*))
 939 FORMAT(/6X,*  FOR ALL TARGETS EXCEPT *,6(2A4,*,*))
 940 FORMAT(/6X,*  FOR ALL TARGETS*)
 941 FORMAT(3A4,2X,A4,I6,I3,A1,I1,1X,5F10.0)
 942 FORMAT(A4,6(1X,A1,I1,1X,2A4))
 944 FORMAT(7X,I2,*  ALL POSSIBLE WEAPON VECTORS WITH *)
 945 FORMAT(12X,A4,A1,I1,* OTHER WEAPONS ARE SELECTED*)
 946 FORMAT(*1*,1X,*INPUT ERROR*)
 947 FORMAT(*1*,57X,*SUMMARY OF RESULTS*///
     *         1X,*PROB OBJ  OBJECTIVE*,7X,*TARGETS*,13X,*VALUE*,13X,
     *           *WEAPONS*,12X,*WEAPONS*,13X,*ENT*,16X,*CMP*/
     *         2X,*NUM NUM    VALUE *,2(5X,*TOTAL      KILL*),
     *           4(5X,*TOTAL      USED*)//)
 948 FORMAT(*0*,I4,I3,F11.2,6(F11.2,F8.2))
 949 FORMAT(10(2I2,F4.0))
C
C  INITIALIZE ROUTINE
C
      N1=0
      NP=NP+1
      CDE=.TRUE.
      ISW=0
      NZ=0
      IF(IFT.NE.0) GO TO 1220
      IFT=1
      DO 1210 I=1,20
      DO 1210 J=1,20
      FTP(I,J)=1.
 1210 CONTINUE
 1220 CONTINUE
C
C  READ START OR I/O CARD
C
 5100 READ(IDSR,914) ICONT,LN,LFLD,TITLE
      IF(EOF,IDSR) 9000,5101
 5101 CONTINUE
      IF(ICONT.EQ.ISTART)GO TO 5300
      IF(ICONT.NE.INOUT )GO TO 5100
C
C  PROCESS I/O CARD
C
      DO 5200 I=1,7
      IOCONT(I)=.FALSE.
      DO 5200 J=1,7
      IF(LFLD(J).EQ.LIO(I))IOCONT(I)=.TRUE.
 5200 CONTINUE
      GO TO 5100
C
C  PROCESS START CARD
C
 5300 IF(LN.EQ.IBLANK)GO TO 5320
```

```
      DO 5310 I=1,10
      IF(LN.EQ.LNUM(I))N1=I
 5310 CONTINUE
      IF(N1.EQ.10)N1=0
      IF(N1.GT.3) GO TO 8000
 5320 JSW=0
      DO 5330 I=1,5
      OBJCON(I)=.FALSE.
      DO 5330 J=1,5
      IF(LFLU(J).NE.LOBJ(I))GO TO 5330
      OBJCON(I)=.TRUE.
      JSW=1
 5330 CONTINUE
      IF(JSW.EQ.0) NZ=1
      IF(JSW.EQ.0)OBJCON(1)=.TRUE.
 5400 READ(IUSR,942)ICONT
C
C  DECODE CONTROL CARD
C
 5410 DO 5420 I=1,9
      IF(ICONT.EQ.LCON(I))GO TO(1000,1010,1007,1017,2100,2100,2000,2050,
     *                         1020),I
 5420 CONTINUE
      IF(ICONT.EQ.4H*FTP) GO TO 1200
      IF(ICONT.EQ.4H*COL) GO TO 1300
      GO TO 5400
C
C  INPUT WEAPON INFORMATION
C
 1000 CDE=.FALSE.
      JSW=1
      NW=0
      I=0
 1001 I=I+1
      READ(IUSR,941)ICONT,(WNAME(J,I),J=1,2),WTYPE(I),NUMW(I),D1,D2,
     *          HOB(I),YLD(I),CEP(I),PLS(I),WSR(I),PTP(I)
      IF(ICONT.NE.IBLANK)GO TO 5410
      NW=I
      IF(NW.GT.20) GO TO 8000
      IF(HOB(I).NE.0)HOB(I)=1
      HOB(I)=HOB(I)+1
      IF(PLS(I).EQ.0.)PLS(I)=1.
      IF(WSR(I).EQ.0.)WSR(I)=1.
      IF(PTP(I).EQ.0.)PTP(I)=1.
      EMT(I)=YLD(I)**.5
      IF(YLD(I).LT.1.)EMT(I)=YLD(I)**(2./3.)
      IF(CEP(I).GT.0.)GO TO 1003
      CMP(I)=9.
      GO TO 1005
 1003 CSQ=1E-6*CEP(I)*CEP(I)
      CMP(I)=YLD(I)**(2./3.)/CSQ
      IF(YLD(I).LT.0.2)CMP(I)=YLD(I)**(.8)/CSQ
      IF(CMP(I).GT.3.)CMP(I)=3.
 1005 GO TO(1001,2010),JSW
C
C  READ NEW WEAPON NUMBERS
C
 1007 READ(IUSR,910)(NUMW(J),J=1,NW)
```

```
      GO TO 5400
C
C  INPUT TARGET INFORMATION
C
 1010 CDE=.FALSE.
      JSW=1
      NT=0
      I=0
 1011 I=I+1
      READ(IUSR,941)ICONT,(TNAME(J,I),J=1,2),ITYPE(I),NUMT(I),VN(I),
     +              PQ(I),AD(I),V(I),DEMIN(I),DEMAX(I),DEAVE(I),DEDLT(I)
      IF(ICONT.NE.IBLANK)GO TO 5410
      NT=I
      IF(NT.GT.20) GO TO 8000
      IF(DEMIN(I).LT.EPS) DEMIN(I)=0.
      IF(DEMAX(I).LT.EPS) DEMAX(I)=0.
      IF(ABS(DEAVE(I)).LT.EPS) DEAVE(I)=0.
      IF(DEDLT(I).LT.EPS) DEDLT(I)=0.
      IF(V(I).LT.EPS) V(I)=1.
      IF(DEMAX(I).EQ.0.)DEMAX(I)=1.
      GO TO(1011,2060),JSW
C
C  READ NEW TARGET NUMBERS
C
 1017 READ(IUSR,910)(NUMT(J),J=1,NT)
      GO TO 5400
C
C  CHANGE OR ADD WEAPON
C
 2000 CDE=.FALSE.
      JSW=2
      I=NW
      GO TO 1001
 2010 K=NW-1
      DO 2020 J=1,K
      IF(WNAME(1,J).EQ.WNAME(1,I) .AND. WNAME(2,J).EQ.WNAME(2,I) .AND.
     +   WTYPE(J).EQ.WTYPE(I))GO TO 2030
 2020 CONTINUE
      GO TO 1001
 2030 NUMW(J)=NUMW(I)
      HOB(J)=HOB(I)
      YLD(J)=YLD(I)
      CEP(J)=CEP(I)
      PLS(J)=PLS(I)
      WSR(J)=WSR(I)
      PTP(J)=PTP(I)
      EMT(J)=EMT(I)
      CMP(J)=CMP(I)
      I=I-1
      NW=NW-1
      GO TO 1001
C
C  CHANGE OR ADD TARGET
C
 2050 CDE=.FALSE.
      JSW=2
      I=NT
      GO TO 1011
```

```
2060 K=NT-1
     DO 2070 J=1,K
     IF(TNAME(1,J).EQ.TNAME(1,I) .AND. TNAME(2,J).EQ.TNAME(2,I) .AND.
    *   TTYPE(J).EQ.TTYPE(I))GO TO 2080
2070 CONTINUE
     GO TO 1011
2080 NUMT(J)=NUMT(I)
     VN(J)=VN(I)
     PQ(J)=PQ(I)
     AD(J)=AD(I)
     V(J)=V(I)
     DEMIN(J)=DEMIN(I)
     DEMAX(J)=DEMAX(I)
     DEAVE(J)=DEAVE(I)
     DEDLT(J)=DEDLT(I)
     I=I-1
     NT=NT-1
     GO TO 1011
C
C  INPUT ADD AND SUB CARDS
C
2100 IC=I-4
     IF(ISW.EQ.1)GO TO 2200
     ISW=1
     DO 2170 K=1,2
     II=NW
     IF(K.EQ.2)II=NT
     NA=0
     DO 2160 L=1,2
     DO 2160 I=1,II
     IF(L.EQ.2)GO TO 2110
     XA=TNAME(1,I)
     XB=TNAME(2,I)
     IF(K.EQ.1) XA=WNAME(1,I)
     IF(K.EQ.1) XB=WNAME(2,I)
     GO TO 2120
2110 XA=TTYPE(I)
     IF(K.EQ.1) XA=WTYPE(I)
     XB=XBLANK
2120 IF(XA.EQ.XBLANK .AND. XB.EQ.XBLANK)GO TO 2160
     IF(NA.EQ.0)GO TO 2140
     IF(I.NE.1 .AND. L.EQ.2 .AND. XA.EQ.TABLE(1,NA,K))GO TO 2150
     DO 2130 J=1,NA
     IF(XA.EQ.TABLE(1,J,K) .AND. XB.EQ.TABLE(2,J,K))GO TO 8000
2130 CONTINUE
2140 NA=NA+1
     TABLE(1,NA,K)=XA
     TABLE(2,NA,K)=XB
     ILB(NA,K)=I
2150 IUB(NA,K)=I
2160 CONTINUE
2170 NENT(K)=NA
2200 JSW=0
     NA=0
     KB=1
     IF(IC.EQ.1)GO TO 2210
     NA=II
     KB=-1
```

```
2210 NA=NA+KB
     KSW=0
2215 READ(IUSR,942)ICONT,(IAB(J),INSW(J,NA),XN(1,J),XN(2,J),J=1,6)
     IF(INSW(6,NA).LT.1) INSW(6,NA)=0
     JA=IC-1
     JB=1-JA
     IF(ICONT.EQ.IFOR)GO TO 2250
     IF(ICONT.EQ.IBLANK)GO TO 2218
     IF(IC.EQ.1)N2=NA-1
     IF(IC.EQ.2)N3=10-NA
     GO TO 2410
2218 IF(KSW.EQ.1)GO TO 2310
     IF(JSW.EQ.1)GO TO 2230
     JSW=1
     DO 2220 I=1,NT
2220 INAP(I,NA)=JB
     GO TO 2310
2230 DO 2240 I=1,NT
2240 INAP(I,NA)=INAP(I,NA-KB)
     GO TO 2310
2250 KSW=1
     JSW=1
     K=1
     IF(XN(1,1).NE.XALL .OR. XN(2,1).NE.XEXC)GO TO 2260
     K=2
     JA=1-JA
     JB=1-JB
2260 DO 2270 I=1,NT
2270 INAP(I,NA)=JA
     DO 2300 I=K,6
     DO 2280 J=1,NTENT
     IF(XN(1,I).EQ.TABLE(1,J,2) .AND.XN(2,I).EQ.TABLE(2,J,2))GO TO 2290
2280 CONTINUE
     GO TO 2215
2290 LA=ILB(J,2)
     LB=IUB(J,2)
     DO 2300 J=LA,LB
2300 INAP(J,NA)=JB
     GO TO 2215
2310 DO 2340 I=1,6
     INRL(I,NA)=1
     IF(IAB(I).EQ.LG .OR. IAB(I).EQ.IGREAT)INRL(I,NA)=2
     IF(IAB(I).EQ.LL .OR. IAB(I).EQ.ILESS )INRL(I,NA)=3
     IF(XN(1,I).EQ.XOTA .AND. XN(2,I).EQ.XOTB)GO TO 2360
     IF(I.EQ.6)GO TO 2350
     DO 2320 J=1,NWENT
     IF(XN(1,I).EQ.TABLE(1,J,1) .AND.XN(2,I).EQ.TABLE(2,J,1))GO TO 2330
2320 CONTINUE
     GO TO 2350
2330 INSC(NA)=I
     INLB(I,NA)=ILB(J,1)
2340 INUB(I,NA)=IUB(J,1)
2350 INRL(6,NA)=4
     GO TO 2210
2360 INSW(6,NA)=INSW(I,NA)
     INRL(6,NA)=INRL(I,NA)
     GO TO 2210
C
```

```
C  INPUT FOOTPRINT EFFICIENCY
C
 1200 CONTINUE
      READ(IUSR,949) (IX(K),JX(K),DJM(K),K=1,10)
      DO 1230 K=1,10
      I=IX(K)
      J=JX(K)
      IF(I.LT.1.OR.J.LT.1) GO TO 5400
      FTP(I,J)=DUM(K)
 1230 CONTINUE
      GO TO 1200
C
C  COLLATERAL DAMAGE FACTOR
C
 1300 CONTINUE
      DO 1320 J=1,NW
      XYZ=25.*YLD(J)
      XYZ=XYZ**0.333
      DO 1310 I=1,NT
      IF(VN(I).GT.15) GO TO 1310
      FTP(I,J)=FTP(I,J)*XYZ
 1310 CONTINUE
 1320 CONTINUE
      GO TO 5400
C
C  OUTPUT WEAPON INFORMATION
C
 1020 WRITE(6,921)NP,TITLE
      WRITE(6,922)
      NWSUM =0
      YLDSUM=0.
      EMTSUM=0.
      CMPSUM=0.
      DO 1025 I=1,NW
      D1=NUMW(I)*YLD(I)
      D2=NUMW(I)*EMT(I)
      D3=NUMW(I)*CMP(I)
      K=LG
      IF(HOB(I).NE.1)K=LAIR
      WRITE(6,915)I,(WNAME(J,I),J=1,2),WTYPE(I),NUMW(I),K,YLD(I),CEP(I),
     *            PLS(I),WSR(I),PTP(I),EMT(I),CMP(I),D1,D2,D3
      NWSUM = NWSUM+NUMW(I)
      YLDSUM=YLDSUM+D1
      EMTSUM=EMTSUM+D2
 1025 CMPSUM=CMPSUM+D3
      WRITE(6,937)NWSUM,YLDSUM,EMTSUM,CMPSUM
C
C  OUTPUT TARGET INFORMATION
C
      WRITE(6,923)
      NTSUM=0
      VSUM=0.
      DO 1030 I=1,NT
      D1=NUMT(I)*V(I)
      WRITE(6,916)I,(TNAME(J,I),J=1,2),TTYPE(I),NUMT(I),VN(I),PQ(I),
     *            AU(I),V(I),D1,DEMIN(I),DEMAX(I),DEAVE(I),DEDLT(I)
      NTSUM=NTSUM+NUMT(I)
 1030 VSUM = VSUM+D1
```

```
      WRITE(6,917)NTSUM,VSUM
C
C  CALCULATE DE TABLE
C
      IF(CDE)GO TO 1038
      DO 1035 I=1,NT
      IT=1
      IF(PQ(I).EQ.XQ)IT=2
      DO 1035 J=1,NW
      PA=PLS(J)*WSR(J)*PTP(J)
      Y=1000.*YLD(J)
      CSAVE=CEP(J)
      CEP(J)=SQRT(CEP(J)*CEP(J)+0.4*6076.*6076.*DEDLT(I)*DEDLT(I))
      CALL DPDX(VN(I),IT,AD(I),Y,CEP(J),HOB(J),PD,DC,DY)
      CEP(J)=CSAVE
      PK(I,J)  =PA*PD
 1035 CONTINUE
C
C  OUTPUT DE TABLE
C
 1038 WRITE(6,924)
      II=-9
 1042 II=II+10
      JJ=II+9
      IF(JJ.GT.NW)JJ=NW
      WRITE(6,925)
      WRITE(6,926)(I,I=II,JJ)
      DO 1045 I=1,NT
 1045 WRITE(6,918)I,(PK(I,J),J=II,JJ)
      IF(JJ.NE.NW)GO TO 1042
C
C  OUTPUT VECTOR SPECS
C
      IF(N2+N3.GT.10) GO TO 8000
      WRITE(6,927)
      WRITE(6,928)N1
      DO 1140 KB=1,2
      KC=NAS(KB)
      IF(KC.EQ.0)GO TO 1140
      IC=3-2*KB
      JB=2-KB
      IF(KB.EQ.1)WRITE(6,929)
      IF(KB.EQ.2)WRITE(6,934)
      DO 1130 I=1,KC
      K=I
      IF(KB.EQ.2)K=11-I
      IF(I.EQ.1)GO TO 1060
      DO 1055 J=1,NT
      IF(INAP(J,K).NE.INAP(J,K-IC))GO TO 1060
 1055 CONTINUE
      GO TO 1100
 1060 DO 1065 J=1,NT
      IF(INAP(J,K).NE.JB)GO TO 1070
 1065 CONTINUE
      WRITE(6,940)
      GO TO 1100
 1070 DO 1075 J=1,NT
      IAP(J,1)=INAP(J,K)
```

```
      IF(KB.EQ.2)IAP(J,1)=1-INAP(J,K)
 1075 IAP(J,2)=1-IAP(J,1)
      DO 1095 II=1,2
      NC(II)=0
      DO 1090 J=1,NTENT
      L=NTENT+1-J
      LA=ILB(L,2)
      LB=IUB(L,2)
      DO 1080 JJ=LA,LB
      IF(IAP(JJ,II).NE.1)GO TO 1090
 1080 CONTINUE
      DO 1085 JJ=LA,LB
 1085 IAP(JJ,II)=0
      NC(II)=NC(II)+1
      NII=NC(II)
      ITB(NII,II)=L
 1090 CONTINUE
      NB(II)=1
      DO 1095 J=1,NT
      IF(IAP(J,II).EQ.1)NB(II)=0
 1095 CONTINUE
      JJ=2
      IF(NB(1).EQ.1 .AND. (NB(2).EQ.0 .OR. NC(1).LE.NC(2)))JJ=1
      II=NC(JJ)
      DO 1097 L=1,II
      IJK=ITB(L,JJ)
      DO 1096 J=1,2
      XTABL(J,L)=TABLE(J,IJK,2)
 1096 CONTINUE
 1097 CONTINUE
      IF(JJ.EQ.1) WRITE(6,938) (XTABL(1,L),XTABL(2,L),L=1,II)
      IF(JJ.EQ.2) WRITE(6,939) (XTABL(1,L),XTABL(2,L),L=1,II)
 1100 DO 1110 J=1,6
      L=INRL(J,K)
      IAB(J)=IBLANK
      IF(L.EQ.2)IAB(J)=IGREAT
 1110 IF(L.EQ.3)IAB(J)=ILESS
      WRITE(6,944)I
      II=IBLANK
      L=INSC(K)
      DO 1120 J=1,L
      DO 1115 JJ=1,NWENT
      IF(INLB(J,K).EQ.ILB(JJ,1) .AND. INUB(J,K).EQ.IUB(JJ,1))GO TO 1117
 1115 CONTINUE
 1117 IF(J.GT.1)II=IAND
      WRITE(6,931)II,IAB(J),INSW(J,K),(TABLE(KK,JJ,1),KK=1,2)
 1120 CONTINUE
      IF(INRL(6,K).EQ.4)GO TO 1130
      WRITE(6,945)IAND,IAB(6),INSW(6,K)
 1130 CONTINUE
 1140 CONTINUE
C
C  BUILD B MATRIX
C
 1900 DO 1910 I=1,NW
 1910 B(I)=NUMW(I)
      M=NW
      DO 1920 I=1,NT
```

```
      M=M+1
 1920 B(M)=NUMT(I)
      J=0
      MZ=M
      DO 1930 I=1,NT
      IF(DEAVE(I).EQ.0.) GO TO 1930
      IF(J.EQ.0) J=I
      IF(DEAVE(I).LT.0.) GO TO 1030
      M=M+1
      IF(M.GT.50) GO TO 8000
      XL=0.
      DO 1925 K=J,I
      XL=XL+V(K)*FLOAT(NUMT(K))
 1925 CONTINUE
      B(M)=-XL*DEAVE(I)
      L=M-MZ
      IF(L.GT.20) GO TO 8000
      JSRT(L)=J
      JEND(L)=I
      J=0
 1930 CONTINUE
      IF(J.NE.0) GO TO 8000
      RETURN
 8000 WRITE(6,946)
      STOP 1
 9000 WRITE(6,947)
      DO 9100 J=1,NSENT
      WRITE(6,948)(ISUM(I,J),I=1,2),(XSUM(I,J),I=1,13)
 9100 CONTINUE
      STOP
      END
```

```
      SUBROUTINE DPDX(VN,T,K1,Y,C,H,P,DPDC,DPDY)
      INTEGER T,H,VN
      DIMENSION A(7,2,2),B(4,2),D(4,10,2)
      DATA A /
     P 8.214,-0.1118, 5.265E-4, 2.162E-5,-6.638E-7, 7.132E-9,-3.064E-11,
     A 8.783,-0.1355, 2.355E-3,-2.080E-4, 9.901E-6,-1.872E-7, 1.227E-09,
     Q 8.315,-0.1033,-7.908E-4,-9.039E-5, 1.438E-5,-5.270E-7, 5.726E-09,
     A 8.789,-0.1120,-6.658E-5,-9.803E-4, 5.833E-5,-1.905E-6, 2.056E-08/
      DATA B /
     P   1.66904   ,  -2.17442   ,   0.260926  ,  -0.0752178,
     Q   1.65946   ,  -2.15466   ,   0.295853  ,  -0.0484718/
      DATA D / 4*0.,
     1  -0.577874  ,   1.56916   ,   0.0019762 ,  -0.0063101,
     2  -1.24391   ,   3.32978   ,  -0.0020688 ,  -0.0469539,
     3  -1.95700   ,   5.35163   ,  -0.0477323 ,  -0.154546 ,
     4  -2.80456   ,   7.74241   ,  -0.225298  ,  -0.345665 ,
     5  -3.81168   ,  10.7222    ,  -0.841780  ,  -0.513504 ,
     6  -5.02081   ,  14.6336    ,  -2.37475   ,  -0.476885 ,
     7  -6.65796   ,  20.1955    ,  -5.98176   ,   0.315062 ,
     8  -8.96601   ,  28.9801    ,  -14.2174   ,   3.11471  ,
     9  -12.7265   ,  45.9954    ,  -35.5644   ,  12.2164   ,
     Q 4*0.,
     1  -0.288917  ,   0.798865  ,  -0.0390065 ,   0.0011575,
     2  -0.611921  ,   1.72876   ,  -0.186010  ,   0.0095419,
     3  -0.978187  ,   2.83679   ,  -0.511458  ,   0.0514480,
     4  -1.40112   ,   4.19642   ,  -1.13390   ,   0.181180 ,
     5  -1.90063   ,   5.91548   ,  -2.23065   ,   0.484749 ,
     6  -2.50907   ,   8.17913   ,  -4.11606   ,   1.12029  ,
     7  -3.28374   ,  11.3172    ,  -7.35613   ,   2.37622  ,
     8  -4.34296   ,  16.0486    ,  -13.2112   ,   4.90666  ,
     9  -6.04000   ,  24.4300    ,  -25.3672   ,  10.6430   /
      IF(C.EQ.0.)GO TO 10
      K=K1+1
      X=Y**(-1./3.)
      V=VN+D(1,K,T)+X*(D(2,K,T)+X*(D(3,K,T)+X*D(4,K,T)))
      DVDX=D(2,K,T)+2*D(3,K,T)*X+3*D(4,K,T)*X*X
      S=EXP(A(1,H,T)+V*(A(2,H,T)+V*(A(3,H,T)+V*(A(4,H,T)+V*(A(5,H,T)+
     +    V*(A(6,H,T)+V* A(7,H,T)))))))
      DSDV=S*(A(2,H,T)+V*(2*A(3,H,T)+V*(3*A(4,H,T)+V*(4*A(5,H,T)+
     +    V*(5*A(6,H,T) +V*6*A(7,H,T))))))
      W=S/X
      U=W/C
      IF(U.GT.10.)GO TO 10
      IF(U.LT.0.1)GO TO 20
      Q=B(1,T)+U*(B(2,T)+U*(B(3,T)+U*B(4,T)))
      P=EXP(-EXP(Q))
      DPDU=-P*EXP(Q)*(B(2,T)+2*B(3,T)*U+3*B(4,T)*U*U)
      DPDC=-U*DPDU/C
      DPDY=DPDU*(W-DSDV*DVDX)/(3*C*Y)
      RETURN
   10 P=1.
      GO TO 30
   20 P=0.
   30 DPDC=0.
      DPDY=0.
      RETURN
      END
```

```
      SUBROUTINE BUILDA
      COMMON /BLKA/ B(51),C(2001),M,N,NZ
      COMMON /BLKB/ NP,NT,NW,NV,TOBJ
      COMMON /BLKF/ N1,N2,N3,INSC(10),INPL(6,10),INSW(6,10),
     *              INLR(5,10),INUB(5,10),INAP(20,10)
      COMMON /BLKG/ FSTENT,MATFUL,VECFND,FINISH,SKPCHK
      COMMON /BLKI/ V(21),DEMIN(21),DEMAX(21),DEAVE(21),DEDLT(21),
     *              YLD(21),EMT(21),CMP(21)
      COMMON /BLKL/ IR(20),IAP(20)
      COMMON /BLKZ/ IA(4,2000),UA(2000),TA(2000),NK
      COMMON /RED / NVW,KA,F
      LOGICAL FSTENT,MATFUL,VECFND,FINISH,SKPCHK
      DO 1001 I=1,8000
 1001 IA(I)=0
      NX=NW+NT
      K=N1+1
      DO 1002 I=1,NT
 1002 IAP(I)=1
      SKPCHK=.TRUE.
      MXN1=0
      MXN2=0
      L=1
 1005 DO 1010 I=L,K
      J=I-1
      NVW=J
      CALL CREATE
 1010 MXN1=J
      SKPCHK=.FALSE.
      IF(N2.EQ.0) GO TO 1170
      DO 1020 I=1,N2
      DO 1017 J=1,NT
 1017 IAP(J)=INAP(J,I)
      KA=I
      CALL KREATE
 1020 MXN2=I
 1170 N=NV+M-NT
      K=NV+1
      L=NV+NW
      DO 1310 J=K,N
 1310 C(J)=0.
      B1=0.
      IF(NZ.EQ.0) GO TO 1330
      DO 1320 J=1,NV
      C(J)=0.
 1320 CONTINUE
      GO TO 1370
 1330 CONTINUE
      NB=NT+1
      NE=NV
      DO 1340 J=NB,NE
      C(J)=-C(J)
 1340 CONTINUE
      DO 1360 I=1,NT
      B1=B1+B(NW+I)*C(I)
      DO 1350 J=NB,NE
      CALL AMAT(NW+I,J,A)
      C(J)=C(J)+A*C(I)
```

```
 1350 CONTINUF
      C(I)=0.
 1360 CONTINUF
 1370 CONTINUE
      B(M+1)=A1
      RETURN
      END
```

```
      FUNCTION OBJECT(IC,IT,P)
      COMMON /BLKB/ NP,NT,NW,NV,IOBJ
      COMMON /BKLD/ PK(20,20),FTP(20,20)
      COMMON /BLKI/ V(21),DEMIN(21),DEMAX(21),DEAVE(21),DEDLT(21),
     *              YLD(21),EMT(21),CMP(21)
      COMMON /BLKL/ IR(20),IAP(20)
      COMMON /RED / NVW,KA,F
      DIMENSION IWP(3)
      DIMENSION OBJ(21,5)
      EQUIVALENCE(OBJ(1,1),DEAVE(1))
      DATA EPS/0.000000001/
      ICNT=0
      P=1.
      DO 60 I=1,NW
      IF(IC.LE.0)IW=IR(I)
      IF(IC.GT.0) CALL AMAT(I,IC,A)
      IF(IC.GT.0) IW=IFIX(A+0.5)
      PS=1.-PK(IT,I)
   50 IF(IW.LE.0)GO TO 60
      P=P*PS
      ICNT=ICNT+1
      IWP(ICNT)=I
      IW=IW-1
      GO TO 50
   60 CONTINUE
      F=1.
      IF(ICNT.EQ.0) GO TO 70
      F=0.
      S=0.
      DO 65 I=1,ICNT
      J=IWP(I)
      F=F+PK(IT,J)*FTP(IT,J)
      S=S+PK(IT,J)
   65 CONTINUE
      F=F/S
   70 CONTINUE
      OBJECT=F*P
C   CHECK MIN AND MAX DE
   75 IF(P.GT.1.-DEMIN(IT)+EPS.AND.P.LT.1.-EPS) GO TO 80
      IF(P.LT.1.-DEMAX(IT)-EPS) GO TO 80
      GO TO(100,200,300,300,300),IOBJ
   80 OBJECT=1001.
      RETURN
C   MAX DE
  100 OBJECT=V(IT)*OBJECT
      RETURN
C   MIN WEAPONS
  200 OBJECT=ICNT
      RETURN
C MIN MEGATONS, EMT, OR CMP
  300 OBJECT=0.
      IF(ICNT.EQ.0)RETURN
      DO 310 I=1,ICNT
      IQ=IWP(I)
  310 OBJECT=OBJECT+OBJ(IQ,IOBJ)
      RETURN
      END
```

```
      SUBROUTINE CREATE
      COMMON /BLKB/ NP,NT,NW,NV,IORJ
      COMMON /BLKF/ N1,N2,N3,INSC(10),INBL(6,10),INSW(6,10),
     *              INLB(5,10),INUB(5,10),INAP(20,10)
      COMMON /BLKG/ FSTENT,MATFUL,VECFND,FINISH,SKPCHK
      COMMON /BLKL/ IRR(20),IAP(20)
      COMMON /RED / NVW,KA,F
      DIMENSION IR(6,20),NSW(6),JL(6),JU(6),ISIZE(6),IFLAG(6)
      LOGICAL FSTENT,MATFUL,VECFND,FINISH,IFLAG
      LOGICAL SKPCHK
C*********************************************************************
C******************* INITIALIZE FOR CREATE ENTRY *********************
C*********************************************************************
      NS=0
      NSS=1
      NSEC=1
      JL(1)=1
      JU(1)=NW
      NSW(1)=NVW
      GO TO 1060
C*********************************************************************
C******************* INITIALIZE FOR KREATE ENTRY *********************
C*********************************************************************
C******************* INITIALIZE FOR KREATE ENTRY *********************
C*********************************************************************
      ENTRY KREATE
      NS=INSC(KA)
      NSS=NS+1
      DO 1000 I=1,NW
 1000 IRR(I)=0
      DO 1005 I=1,NS
      IA=INLB(I,KA)
      IB=INUB(I,KA)
      DO 1002 J=IA,IB
 1002 IRR(J)=1
 1005 NSW(I)=INSW(I,KA)
      NSW(NSS)=INSW(6,KA)
      NSEC=0
      ISR=0
      DO 1040 I=1,NW
      IF(ISR.EQ.1)GO TO 1020
      IF(IRR(I).NE.0)GO TO 1040
      NSEC=NSEC+1
      JL(NSEC)=I
      ISR=1
 1020 JU(NSEC)=I
      IF(IRR(I).NE.1)GO TO 1040
      JU(NSEC)=I-1
      ISR=0
 1040 CONTINUE
      IF(NSEC.EQ.0)NSS=NS
C*********************************************************************
C****************** CODE COMMON TO CREATE AND KREATE *****************
C*********************************************************************
C
C  CALCULATE ISIZE AND INITIALIZE IR ARRAYS
C
```

```
1060 DO 1110 J=1,NSS
     IF(J.EQ.NSS .AND. NSEC.GT.0)GO TO 1070
     ISIZE(J)=INUB(J,KA)-INLB(J,KA)+1
     GO TO 1090
1070 ISIZE(NSS)=0
     DO 1080 K=1,NSEC
1080 ISIZE(NSS)=ISIZE(NSS)+JU(K)-JL(K)+1
1090 CONTINUE
     IQ=ISIZE(J)
     IR(J,IQ)=NSW(J)
     IF(ISIZE(J).EQ.1)GO TO 1110
     L=ISIZE(J)-1
     DO 1100 K=1,L
1100 IR(J,K)=0
1110 IFLAG(J)= .TRUE.
C
C  TRANSFER IR SECTIONS TO IRR IF IR SECTION HAS CHANGED (IFLAG(I)=TRUE)
C
1200 IF(NS.EQ.0)GO TO 1220
     DO 1215 I=1,NS
     IF(.NOT.IFLAG(I))GO TO 1215
     IFLAG(I)=.FALSE.
     K=ISIZE(I)
     L=INLB(I,KA)-1
     DO 1210 J=1,K
     L=L+1
1210 IRR(L)=IR(I,J)
1215 CONTINUE
C
C  TRANSFER IR(NSS) TO IRR IF IT HAS BEEN CHANGED (IFLAG(NSS)=TRUE)
C
1220 IF(.NOT.IFLAG(NSS) .OR. NSEC.EQ.0)GO TO 1235
     IFLAG(NSS)=.FALSE.
     K=0
     DO 1230 I=1,NSEC
     IA=JL(I)
     IB=JU(I)
     DO 1230 J=IA,IB
     K=K+1
1230 IRR(J)=IR(NSS,K)
C
C  CALL INSERT AND RETURN 1 IF THE A MATRIX IS FULL
C
1235 CALL INSERT
C
C  BUMP ONE WPN DOWN ONE PLACE .IF ALL WPNS ARE AT BOTTOM GOTO 1260
C
     I=1
1240 IF(IR(I,1).EQ.NSW(I)) GO TO 1260
     IFLAG(I)=.TRUE.
     J=1
1250 J=J+1
     IF(IR(I,J).LT.1)GO TO 1250
     IR(I,J)=IR(I,J)-1
     K=IR(I,1)+1
     IR(I,1)=0
     IR(I,J-1)=K
     GO TO 1200
```

```
C
C  RFINITIALIZE SECTION I. IF I=NSS RETURN ELSE INCREMENT I
C
 1260 IFLAG(I)=.TRUE.
      IR(I,1)=0
      IQ=ISIZE(I)
      IR(I,IQ)=NSW(I)
      I=I+1
      IF(I.LE.NSS)GO TO 1240
      RETURN
      END
```

```
      SUBROUTINE INSERT
      COMMON /BLKA/ B(51),C(2001),M,N,N7
      COMMON /BLKB/ NP,NT,NW,NV,TOBJ
      COMMON /BLKF/ N1,N2,N3,INSC(10),INRL(6,10),INSW(6,10),
     *              INLB(5,10),INUB(5,10),INAP(20,10)
      COMMON /BLKG/ FSTENT,MATFUL,VECFND,FINISH,SKPCHK
      COMMON /BLKI/ V(21),DEMIN(21),DEMAX(21),DEAVE(21),DEDLT(21),
     *              YLD(21),EMT(21),CMP(21)
      COMMON /BLKL/ IRR(20),IAP(20)
      COMMON /BLKZ/ IA(4,2000),UA(2000),TA(2000),NX
      COMMON /RED / NVW,KA,F
      DIMENSION IN(20),IC(20)
      LOGICAL FSTENT,MATFUL,VECFND,FINISH,SKPCHK
      DO 5 I=1,NT
    5 IN(I)=IAP(I)
      IF(N3.EQ.0) GO TO 100
      DO 50 I=1,N3
      K=11-I
      DO 6 J=1,NW
    6 IC(J)=1
      KS=INSC(K)+1
      DO 40 J=1,KS
      KX=0
      IF(J.EQ.KS)GO TO 8
      KR=INRL(J,K)
      KN=INSW(J,K)
      KL=INLB(J,K)
      KU=INUB(J,K)
      DO 7 L=KL,KU
      KX=KX+IRR(L)
    7 IC(L)=0
      GO TO(10,20,30,40),KR
    8 KR=INRL(6,K)
      KN=INSW(6,K)
      DO 9 L=1,NW
    9 KX=KX+IC(L)*IRR(L)
      GO TO (10,20,30,40),KR
   10 IF(KX.NE.KN)GO TO 50
      GO TO 40
   20 IF(KX.LE.KN)GO TO 50
      GO TO 40
   30 IF(KX.GE.KN)GO TO 50
   40 CONTINUE
      ISW=0
      DO 45 J=1,NT
      IN(J)=IN(J)*INAP(J,K)
      IF(IN(J).NE.0)ISW=1
   45 CONTINUE
      IF(ISW.EQ.0)RETURN
   50 CONTINUE
  100 IF(NV.EQ.0 .OR. SKPCHK)GO TO 220
      DO 120 I=1,NV
      DO 110 J=1,NW
      CALL AMAT(J,I,A)
      LA=IFIX(A+0.5)
      IF(IRR(J).NE.LA) GO TO 120
  110 CONTINUE
```

```
      J=NW
115   J=J+1
      CALL AMAT(J,I,A)
      IF(A.EW.0.) GU TO 115
      IN(J-NW)=0
120   CONTINUE
220   DO 250 I=1,NT
      IF(IN(I).EQ.0)GO TO 250
      C(NV+1)=OBJECT(0,I,Y)
      IF(C(NV+1).GT.1000.) GO TU 250
      NV=NV+1
      IA(1,NV)=I
      DA(NV)=V(I)*(Y-1.)*F
      TA(NV)=F
      K=1
      DO 240 J=1,NW
      IF(IRR(J).EQ.0) GO TO 240
      IF(IRR(J).GT.1) GO TO 230
      K=K+1
      IA(K,NV)=J
      GO TO 240
230   CONTINUE
      IF(IRR(J).EQ.2) IA(4,NV)=J
      IF(IRR(J).EQ.3) IA(3,NV)=J
240   CONTINUE
250   CONTINUE
      IF(NV+M-NT.LT.2000) RETURN
      WRITE(9,1001)
      STOP 2
1001  FORMAT(*1  WEAPON VECTOR LTMIT EXCFEDED*)
      END
```

```
      SUBROUTINE SIMPLE
      COMMON /BLKA/ B(51),C(2001),M,N,NZ
      COMMON /BLKB/ NP,NT,NW,NV,IOB
      COMMON /BLKC/ JA(51),XA(51)
      COMMON /BLUE/ Q(51,51),M1
      DIMENSION D(51),E(51)
      DATA EPS/0.000000001/
C
C  INITIALIZATION
C
      IFLAG=0
      M1=M+1
      NE=NV
      NN=N+NZ
      N1=NN+1
      DO 20 I=1,M1
      DO 10 J=1,M1
   10 Q(I,J)=0.
      D(I)=B(I)
      Q(I,I)=1.
   20 CONTINUE
      IF(NZ.EQ.0) GO TO 40
      DO 30 I=1,NW
   30 D(I)=0.
   40 CONTINUE
      DO 50 I=1,NW
   50 JA(I)=NE+I
      DO 60 I=1,NT
      I1=NW+I
      I2=NW+NT+I
      JA(I1)=I
      IF(I2.LT.M1) JA(I2)=NE+NW+I
   60 CONTINUE
      JA(M1)=N1
      DO 70 I=1,M1
   70 XA(I)=D(I)
C
C  ITERATION
C
  100 CONTINUE
      DO 110 I=1,M
      IF(D(I).LT.0.) GO TO 120
  110 CONTINUE
      IFLAG=1
      GO TO 200
  120 CONTINUE
      IT=I
      JS=0
      XT=0.
      DO 130 J=1,NN
      CALL AX(IT,J,TT)
      IF(TT.GT.XT-EPS) GO TO 130
      XT=TT
      JS=J
  130 CONTINUE
      IF(JS.EQ.0) GO TO 500
      IS=0
```

```
      YT=-1.
      DO 140 I=1,M
      CALL AX(I,JS,TT)
      E(I)=TT
      IF(TT.LT.EPS) GO TO 140
      IF(D(I).LT.0.) GO TO 140
      ZT=D(I)/TT
      IF(ZT.GT.YT.AND.YT.GT.-0.5) GO TO 140
      YT=ZT
      IS=I
  140 CONTINUE
      IF(IS.EQ.0) IS=IT
      CALL AX(M1,JS,TT)
      E(M1)=TT
      GO TO 300
  200 CONTINUE
      JS=0
      XT=0.
      DO 210 J=1,NN
      CALL AX(M1,J,TT)
      IF(TT.LT.XT+EPS) GO TO 210
      XT=TT
      JS=J
  210 CONTINUE
      IF(JS.EQ.0) GO TO 400
      E(M1)=XT
      IS=0
      YT=-1.
      DO 220 I=1,M
      CALL AX(I,JS,TT)
      E(I)=TT
      IF(TT.LT.EPS) GO TO 220
      ZT=D(I)/TT
      IF(ZT.GT.YT.AND.YT.GT.-0.5) GO TO 220
      YT=ZT
      IS=I
  220 CONTINUE
      IF(IS.EQ.0) GO TO 600
  300 CONTINUE
      JA(IS)=JS
      TT=E(IS)
      DO 310 I=1,M1
      E(I)=-E(I)/TT
  310 CONTINUE
      E(IS)=1./TT
      DO 330 I=1,M1
      IF(I.EQ.IS) GO TO 330
      D(I)=D(I)+E(I)*D(IS)
      DO 320 J=1,M1
      Q(I,J)=Q(I,J)+E(I)*Q(IS,J)
  320 CONTINUE
  330 CONTINUE
      D(IS)=E(IS)*D(IS)
      DO 340 J=1,M1
      Q(IS,J)=E(IS)*Q(IS,J)
  340 CONTINUE
      DO 350 I=1,M1
      XA(I)=D(I)
```

```
  350 CONTINUE
      IF(IFLAG.EQ.1) GO TO 200
      GO TO 100
C
C  SOLUTION
C
  400 CONTINUE
      IF(D(M1).GE.0.) RETURN
  500 WRITE(6,1001)
      GO TO 700
  600 WRITE(6,1002)
  700 CONTINUE
      STOP 3
 1001 FORMAT(*)  PROBLEM INFEASIBLE*)
 1002 FORMAT(*)  PROBLEM UNBOUNDED*)
      END
```

```
      SUBROUTINE OUTMAT
      COMMON /BLKA/ B(51),C(2001),M,N,NZ
      COMMON /BLKB/ NP,NT,NW,NV,TOBJ
      DIMENSION IN(50),IM(50),MA(50)
      LC=0
      DO 110 I=1,N
      IF(MOD(LC,56).EQ.0)WRITE(6,900)(J,J=1,N),(J,J=1,NT)
      LC=LC+1
      DO 100 J=1,M
      CALL AMAT(J,I,A)
      LA=IFIK(A+0.5)
      MA(J)=IABS(LA)
100   CONTINUE
110   WRITE(6,910) I,C(I),(MA(J),J=1,M)
      DO 120 I=1,M
120   IN(I)=ABS(B(I))+0.1
      WRITE(6,920)
      J=0
      K=1
130   J=J+1
      K=10*K
      DO 140 I=1,M
      IF(K.LE.IN(I))GO TO 130
140   CONTINUE
      DO 160 L=1,J
      K=K/10
      DO 150 I=1,M
150   IM(I)=MOD(IN(I)/K,10)
160   WRITE(6,930)(IM(I),I=1,M)
      RETURN
900   FORMAT (*1*,8X,*C*,3X,*A MATRIA*//12X,60I2)
910   FORMAT (1X,I3,F7.2,1X,60I2)
920   FORMAT (/7X,*B =  *)
930   FORMAT (12X,60I2)
      END
```

```
      SUBROUTINE OUTPUT
      COMMON /BLKA/ B(51),C(2001),M,N,NZ
      COMMON /BLKB/ NP,NT,NW,NTV,IO3U
      COMMON /BLKC/ JA(51),XA(51)
      COMMON /BKLD/ PK(20,20),FTP(20,20)
      COMMON /BLKE/ PUN,DSC,DIN,MAT,VEC,NOL,NUS,TITLE(10)
      COMMON /BLKH/ NTSUM,NWSUM,VSUM,YLDSUM,EMTSUM,CMPSUM
      COMMON /BLKI/ V(21),DEMIN(21),DEMAX(21),DEAVE(21),DEDLT(21),
     *              YLD(21),EMT(21),CMP(21)
      COMMON /BLKJ/ WNAME(2,21),TNAME(2,21),WTYPE(21),TTYPE(21),
     *              NUMW(21),NUMT(21),VN(21),PW(21),AD(21),CEP(21),
     *              PLS(21),WSR(21),PTP(21),HOB(21)
      COMMON /BLKN/ NSENT,ISUM(2,20),XSUM(13,20)
      COMMON /BLKZ/ IA(4,2000),DA(2000),TA(2000),NX
      DIMENSION HEAD(5,5)
      DIMENSION XDE(50),XWPN(50),XYLD(50),XEMT(50),XCMP(50)
      DIMENSION JTGT(50),LINK(50),IPT(50),IW(20),W(20),MA(20)
      INTEGER HOB,VN,AD
      LOGICAL PUN,USC,DIN,MAT,VEC,NOL,NOS
      DATA HEAD/2HAX,4HVALU,4HE DA,4HMAGE,1HD,2HIN,4HWEAP,4HONS ,4HUSED,
     1          1H ,2HIN,4HMEGA,4HTONS,4H USE,1HD,2HIN,4HEMT ,4HUSED,
     2          1H ,1H ,2HIN,4HCMP ,4HUSED,1H ,1H /
  900 FORMAT(41X,*PROBLEM*,I4,* OPTIMAL SOLUTION FOR MA,A2,*IMUM *,4A4/
     *       41X,10A4/)
  902 FORMAT(//59X,*TARGET RESULTS*//
     1       6X,*TARGET   TYPE         VALUE        TARGETS DAMAGED*,
     2       7X,*WEAPONS USED*,7X,*MEGATONS USED*,10X,*EMT USED*,
     3       12X,*CMP USED*/
     4       24X,*DAMAGED *,5(4X,*NUMBER   PERCENT*)/)
  903 FORMAT(1X,I2,2X,2A4,2X,A4,F12.2,5(F12.2,F8.2))
  904 FORMAT(/5X,*TOTALS*,F20.2,5(F12.2,F8.2)/19X,F12.2,* *)
  905 FORMAT(///60X,*WEAPON RESULTS*//
     *       32X,*WEAPON   TYPE*,8X,*WEAPONS USED       WEAPONS REMAINING*
     *       ,4X,*       */46X,2(5X,*NUMBER   PERCENT*),4X,*      */)
  906 FORMAT(27X,I2,2X,2A4,2X,A4,2(F13.2,F8.2),F10.4)
  907 FORMAT(/31X,*TOTALS*,8X,2(F13.2,F8.2))
  908 FORMAT(*1*,9X,*LAYDOWN*//10X,*NUMBER      DE        WEAPONS*/27X,
     *       26I4/27X,14I4)
  909 FORMAT(*0*,   *TARGET*,I3,2X,2A4,2X,A4/)
  910 FORMAT(9X,F8.2,F8.4,2X,26I4/27X,14I4)
  911 FORMAT(/1X,*TOTALS=*,F9.2,F8.4,2X,26I4/27X,14I4)
  924 FORMAT(20I4)
  925 FORMAT(8F10.5)
  926 FORMAT(I2,F8.2,F10.6,4012)
  930 FORMAT(*1*)
  931 FORMAT(*1*,39X,*SENSITIVITY OF WEAPON SYSTEM PARAMETERS*//6X,
     *       *WEAPON   TYPE        SHADOW        DDE/DCEP          DDE/DYLD*,7X,
     *       *DDE/DPLS          DDE/DWSR        DDE/DPTP          DDE/DPA*/
     *       24X,*PRICE*/)
  932 FORMAT(1X,I2,2X,2A4,2X,A4,F11.5,6E15.4)
  950 FORMAT(*1  THE RELATIVE FORCE SIZE IS*,F6.2)
C
      IF(NZ.EQ.0) GO TO 1500
      R=XA(M+1)
      R=1./R
      WRITE(6,950)R
 1500 CONTINUE
```

```
      WRITE(6,930)
C
C  INITIALIZE ROUTINE
C
  999 ZWPN=0
      ZYLD=0.
      ZEMT=0.
      ZCMP=0.
      ZTK =0.
      ZVK =0.
      ICNT=0
      DO 1300 I=1,NW
      W(I)=0.
 1300 CONTINUE
      DO 1320 J=1,M
      JJA=JA(J)
      IF(JJA.GT.NTV) GO TO 1320
      DO 1310 I=1,NW
      CALL AMAT(I,JJA,A)
      IF(A.EQ.0.) GO TO 1310
      W(I)=W(I)+A*XA(J)
 1310 CONTINUE
 1320 CONTINUE
C
C  INTERPRET SOLUTION JA,XA
C
      DO 1007 I=1,M
C
C  IF SLACK VARIABLE GO TO 1005
C
      IF(JA(I).GT.NTV)GO TO 1005
      IF(XA(I).GT.0.)ICNT=ICNT+1
      JQ=JA(I)
C
C  FIND TARGET NUMBER
C
      J=0
 1001 J=J+1
      NQ=NW+J
      CALL AMAT(NQ,JQ,A)
      IF(A.EQ.0.) GO TO 1001
      JTGT(I)=J
C
C  CALCULATE DE
C
      Y=OBJECT(JA(I),J,PS)
      XDE(I)=(1.-PS)*TA(JQ)
C
C  INITIALIZE VECTOR PARAMETERS
C
      XWPN(I)=0.
      XYLD(I)=0.
      XEMT(I)=0.
      XCMP(I)=0.
C
C  CALCULATE VALUE OF VECTOR PARAMETERS
C
      DO 1002 J=1,NW
```

```
      CALL AMAT(J,JQ,A)
      IF(A.EQ.0) GO TO 1002
      XWPN(I)=XWPN(I)+A
      XYLD(I)=XYLD(I)+A*YLD(J)
      XEMT(I)=XEMT(I)+A*EMT(J)
      XCMP(I)=XCMP(I)+A*CMP(J)
 1002 CONTINUE
C
C  SUM SOLUTION VALUES
C
      ZWPN=ZWPN+XA(I)*XWPN(I)
      ZYLD=ZYLD+XA(I)*XYLD(I)
      ZEMT=ZEMT+XA(I)*XEMT(I)
      ZCMP=ZCMP+XA(I)*XCMP(I)
      ZTK =ZIK +XA(I)*XDE(I)
      JJ=JTGT(I)
      ZVK=ZVK+XA(I)*XDE(I)*V(JJ)
      GO TO 1007
C
C  IF SLACK CORRESPONDS TO A WEAPUN, PUT NEG WPN NUM IN JT3T
C
 1005 JTGT(I)=NTV-JA(I)
      IF(JTGT(I).LT.-NW) JTGT(I)=0
 1007 CONTINUE
C
C  LINK TARGETS
C
      DO 1020 I=1,NT
      JPT=0
      DO 1010 J=1,M
      IF(JTGT(J).NE.I)GO TO 1010
      IF(JPT.EQ.0)IPT(I)=J
      IF(JPT.NE.0)LINK(JPT)=J
      JPT=J
 1010 CONTINUE
 1020 LINK(JPT)=0
      WRITE(6,900)NP,(HEAD(J,IOW)),J=1,5),TITLE
      WRITE(6,902)
      DO 1040 I=1,NT
      YTK =0.
      YWPN=0.
      YYLD=0.
      YEMT=0.
      YCMP=0.
      J=IPT(I)
 1030 YTK =YIK +XA(J)* XDE(J)
      YWPN=YWPN+XA(J)*XWPN(J)
      YYLD=YYLD+XA(J)*XYLD(J)
      YEMT=YEMT+XA(J)*XEMT(J)
      YCMP=YCMP+XA(J)*XCMP(J)
      J=LINK(J)
      IF(J.NE.0)GO TO 1030
      YVK=YTK*V(I)
      IF(NUMI(I).EQ.0)P1=100.
      IF(NUMI(I).GT.0)P1=100.*YTK/NJMT(I)
      P2=100.*YWPN/ NWSUM
      P3=100.*YYLD/YLDSUM
      P4=100.*YEMT/EMTSUM
```

```
      P5=100.*YCMP/CMPSUM
 1040 WRITE(6,903)I,(TNAME(J,I),J=1,2),TTYPE(I),YVK,YTK,P1,YWPN,P2,YYLD,
     *            P3,YEMT,P4,YCMP,P5
      P1=100.*ZTK / NTSUM
      P2=100.*ZWPN/ NWSUM
      P3=100.*ZYLD/YLDSUM
      P4=100.*ZEMT/EMTSUM
      P5=100.*ZCMP/CMPSUM
      P6=100.*ZVK/VSUM
      WRITE(6,904)ZVK,ZTK,P1,ZWPN,P2,ZYLD,P3,ZEMT,P4,ZCMP,P5,P6
C
C  FILL SUMMARY ARRAYS
C
      NSENT=NSENT+1
      IF(NSENT.GT.20) NSENT=20
      ISUM( 1,NSENT)=NP
      ISUM( 2,NSENT)=IOBJ
      GO TO(1042,1043,1044,1045,1046),IOBJ
 1042 XSUM( 1,NSENT)=ZVK
      GO TO 1047
 1043 XSUM( 1,NSENT)=ZWPN
      GO TO 1047
 1044 XSUM( 1,NSENT)=ZYLD
      GO TO 1047
 1045 XSUM( 1,NSENT)=ZEMT
      GO TO 1047
 1046 XSUM( 1,NSENT)=ZCMP
 1047 XSUM( 2,NSENT)= NTSUM
      XSUM( 3,NSENT)=P1
      XSUM( 4,NSENT)=  VSUM
      XSUM( 5,NSENT)=P6
      XSUM( 6,NSENT)= NWSUM
      XSUM( 7,NSENT)=P2
      XSUM( 8,NSENT)=YLDSUM
      XSUM( 9,NSENT)=P3
      XSUM(10,NSENT)=EMTSUM
      XSUM(11,NSENT)=P4
      XSUM(12,NSENT)=CMPSUM
      XSUM(13,NSENT)=P5
      WRITE(6,905)
      DO 1050 I=1,NW
      W(I)=NUMW(I)-W(I)
      J=NTV+I
      P1=NUMW(I)-W(I)
      IF(NUMW(I).EQ.0)P2=100.
      IF(NUMW(I).NE.0)P2=100.*P1/NUMW(I)
      P3=100.-P2
      WRITE(6,906)I,(WNAME(K,I),K=1,2),WTYPE(I),P1,P2,W(I),P3
 1050 CONTINUE
      P1=100.*ZWPN/NWSUM
      P2=NWSUM-ZWPN
      P3=100.-P1
      WRITE(6,907)ZWPN,P1,P2,P3
      IF(NOL)RETURN
      WRITE(6,908)(J,J=1,NW)
      DO 1110 I=1,NT
      WRITE(6,909)I,(TNAME(J,I),J=1,2),TTYPE(I)
      DO 1060 J=1,NW
```

```
1060 W(J)=0.
     YDE=0.
     K=IPT(I)
1070 IF(XA(K).EQ.0.)GO TO 1090
     JQ=JA(K)
     DO 1075 J=1,NW
     CALL AMAT(J,JQ,A)
     MA(J)=IFIX(A+0.5)
1075 CONTINUE
     WRITE(6,910) XA(K),XDE(K),(MA(J),J=1,NW)
     YDE=YDE+XA(K)*XDE(K)
     DO 1080 J=1,NW
     CALL AMAT(J,JQ,A)
     W(J)=W(J)+A*XA(K)
1080 CONTINUE
1090 K=LINK(K)
     IF(K.GT.0)GO TO 1070
     IF(NUMT(I).EQ.0)YDE=1.
     IF(NUMT(I).GT.0)YDE=YDE/NUMT(I)
     DO 1100 J=1,NW
     IW(J)=IFIX(W(J)+0.5)
1100 CONTINUE
     P1=NUMT(I)
1110 WRITE(6,911)P1,YDE,(IW(J),J=1,NW)
     RETURN
     END
```

```
      SUBROUTINE DUAL
      COMMON /BLKA/ B(51),C(2001),M,N,NZ
      COMMON /BLKC/ JA(51),XA(51)
      WRITE(6,900)
      M1=M+1
      DO 110 I=1,M1
  110 WRITE(6,904)I,JA(I),XA(I)
      RETURN
  900 FORMAT(1H1)
  904 FORMAT(1X,2I6,F15.6)
      END
```

```
      SUBROUTINE AMAT(I,J,A)
      COMMON /BLKA/ B(51),C(2001),M,N,NZ
      COMMON /BLKB/ NP,NT,NW,NV,TOR
      COMMON /BLKY/ JSRT(20),JEND(20)
      COMMON /BLKZ/ IA(4,2000),DA(2000),TA(2000),NX
      A=0.
      IF(I.LE.NX.AND.J.LE.NV) GO TO 100
      IF(J.GT.N) GO TO 20
      IF(I.GT.M) GO TO 10
      IF(I.GT.NX.AND.J.LE.NV) GO TO 40
      IF(I.GT.NX) GO TO 50
      IF(I.GT.NW) RETURN
      IF(I.EQ.J-NV) A=1.
      RETURN
   10 CONTINUE
      A=C(J)
      RETURN
   20 CONTINUE
      IF(J.NE.N+NZ+1) GO TO 30
      IF(I.GT.M) A=1.
      RETURN
   30 CONTINUE
      IF(I.GT.M) A=-1.
      IF(I.LE.NW) A=-B(I)
      RETURN
   40 CONTINUE
      IF(IA(1,J).GE.JSRT(I-NX).AND.IA(1,J).LE.JEND(I-NX)) A=DA(J)
      RETURN
   50 CONTINUE
      IF(I.EQ.J-NV+NT) A=1.
      RETURN
  100 CONTINUE
      IF(I.LE.NW) GO TO 200
      IF(I.EQ.NW+IA(1,J)) A=TA(J)
      RETURN
  200 CONTINUE
      IF(I.EQ.IA(4,J)) GO TO 300
      IF(I.EQ.IA(2,J).OR.I.EQ.IA(3,J)) A=1.
      IF(I.EQ.IA(3,J).AND.IA(2,J).EQ.0) A=3.
      RETURN
  300 CONTINUE
      A=1.
      IF(IA(3,J).EQ.0) A=2
      RETURN
      END
```

```
      SUBROUTINE AX(I,J,TT)
      COMMON /BLUE/ Q(51,51),M1
      TT=0.
      DO 10 K=1,M1
      CALL AMAT(K,J,A)
      TT=TT+Q(I,K)*A
   10 CONTINUE
      RETURN
      END
```